

REMARKS

Claims 3 and 11 were previously cancelled without prejudice and claims 1, 2, 4-6, 10, 12-15, and 23-28 were previously withdrawn. Claims 7-9 and 16-22 remain pending in the application, for a total of ten (10) pending claims. Claims 7 and 8 are independent claims, while claims 9 and 16-22 are dependent claims.

The Patent Office previously rejected all of the pending claims (claims 7-9 and 16-22) under 35 U.S.C. § 102(b) based on public use or sale citing the Information Disclosure Statement (IDS) Applicant filed on 4/5/2002. The Patent Office argued that because the Applicant received payment for accounting services performed prior to the critical date, that an offer for sale occurred prior to the critical date and that the invention is bared from patentability. In response, Applicant has stated and provided evidence that the activities that took place prior to the critical date were limited to permissible experimental use. In the Advisory Action mailed 12/20/05, the Patent Office indicated that the Applicant had overcome this rejection, apparently accepting the validity of the evidence provided and concluding that activities more than one year before the filing date were limited to permissible experimental use.

In the Final Office Action mailed 10/27/05, the Patent Office also rejected all of the pending claims (claims 7-9 and 16-22) under 35 U.S.C. § 103 as being unpatentable (obvious) over Blasnik *et al.* (US 2003/0050877). The Patent Office stated that Blasnik taught everything in the claims except for the "sequence of the method steps", which "would have been obvious to one of ordinary skill in the art at the time of the invention". But in the Request for Reconsideration mailed on 11/22/05 in response to the Final Office Action, Applicant submitted that Blasnik *et al.* does not qualify as prior art against Applicant's invention. Rather, Blasnik was filed on September 10, 2001, and as stated in the Green Affidavit previously filed on 8/15/05, "the invention was used by the Bank of New York (BNY) to perform accounting services for JP Morgan prior to February 15, 2001", about seven months prior to the filing of Blasnik. Applicants information disclosure statement (IDS) filed on 4/5/2002, also indicated that the present invention was used between January 31, 2001, and the

summer or fall of 2001. Thus, undisputed evidence had previously been placed on that record that the Applicant invented the current invention prior to the filing of Blasnik *et al.* Further, this evidence was accepted by the Patent Office to show use before the bar date, and then was also accepted by the Patent Office to show that the use before the bar date was limited to permissible experimental use. Still further, Applicant submitted that Applicant was diligent in the pursuit and reduction to practice of the invention and that the experimentation performed on the invention, as described in the Green Affidavit, constituted undisputed evidence of such diligence.

Then, in the Advisory Action mailed on 12/20/05, the Patent Office stated that the Request for Reconsideration mailed on 11/22/05 did not place the application in a condition for allowance because the Applicant had not provided drawing exhibits or records (or explained their absence) to establish reduction to practice or conception of the invention prior to the effective date of Blasnik as required by 37 C.F.R. 1.131(b). Further, such documents had not bee presented to establish due diligence to reduce the invention to practice or to file the application.

Responsive thereto, and in an effort to resolve this final issue, Applicant submits herewith for consideration, the following documents:

1. An e-mail sent on 03/02/2001 from Spencer Moser of the Bank of New York (BNY, the assignee) to Michael Povman (also of BNY) and attachments to that e-mail, which include text and figures describing the BNY system that the current patent application concerns.

2. A fax cover sheet dated May 8, 2001 which transmitted an earlier draft copy of the current patent application from Applicant's patent attorney to Kirk Woetzel (who's actual name is Kurt Woetzel) of BNY for review prior to filing of the current patent application.

Applicant submits that these documents (1) are true copies kept in the regular course of business and that to Applicant's knowledge and belief, the dates on these documents are correct. However, the applications handling these documents has

changed since 2001. As a result, certain figures did not print correctly (e.g., on pages 27-29) of the enclosure titled "The Bank of New York Strategic Global Accounting System Architecture". Applicant's counsel tried several times to get these figures to print but was not successful. But Applicant faxed these three pages to Applicant's counsel, and the faxed pages are included herewith in addition to the pages that did not print correctly. Further, the undersigned submits that the fax cover sheet (2) is a true copy kept in the regular course of business and further that in the regular course of business the draft patent application that had been sent with the May 8, 2001 fax cover sheet was not kept, but rather, only the final draft of the patent application that was filed with the Patent Office was kept in the attorney files.

Both of these dated documents predate the September 10, 2001 filing date and priority date of Blasnik. Further, these documents support the Green Affidavit previously submitted on 8/15/05, and the April 5, 2002 IDS. These documents, separately or combined, provide undisputed evidence that the present invention was conceived and reduced to practice prior to the filing of the Blasnik application.

Still further, applicant submits that they were diligent in the pursuit of the invention including reduction to practice, experimental use, and patenting. It is true that the system that the invention concerns is a large project and that it took time to perform these activities. But as stated in the Green Affidavit submitted on 8/15/05, changes were made to the software during experimental use, and use prior to the critical date was performed for no longer than necessary before the determination was made that the invention worked satisfactorily. In addition, it was in BNY's best interest to get the invention into use as quickly as possible, and it was in BNY's best interest to get a patent application filed as soon as possible to permit them to stop others from copying the invention. As a result, the necessary work was performed with a fully reasonable level of diligence and as quickly as practicable under the circumstances. Accordingly, Applicant was diligent in the pursuit of the invention from the time of conception through filing of the patent application.

In conclusion, Applicant has now submitted documentary evidence from before the priority date of Blasnik indicating that the present invention was conceived and reduced to practice before the priority date of Blasnik. Further, Applicant has explained why other documentation was not kept, and Applicant has attested and explained that they were diligent in reducing the invention to practice, conducting experimental use, and filing of the patent application. Consequently, Applicant submits that Blasnik *et al.* does not qualify as prior art under 35 U.S.C. § 103, and should be eliminated from consideration as a reference. Applicant further submits that the other references cited do not, at least without Blasnik, teach or suggest all of the limitations of the current claims. As a result, Applicant requests reconsideration of this rejection under 35 U.S.C. § 103. Further, Applicant submits that all grounds for rejection have been overcome. Reconsideration and allowance of all pending claims is requested.

Respectfully submitted,

Date: 1-27-06

By:

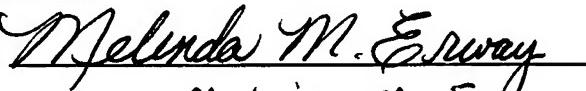


Allan W. Watts
U.S. Reg. No. 45,930
Bryan Cave LLP
One Renaissance Square
Two North Central Avenue, Suite 2200
Phoenix, AZ 85004-4406
allan.watts@bryancave.com
Direct: 602-364-7331
Fax: 602-716-8331

CERTIFICATE OF EXPRESS MAILING UNDER 37 C.F.R. 1.10.

I hereby certify that this document (and any others referred to as being attached or enclosed) is being deposited with the United States Postal Service as "Express Mail Post Office to Addressee" service, mailing label No. **EV478774021US** on **January 27, 2006** and addressed to Mail Stop Amendment, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code, and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.



Printed Name: Melinda M. Erway

1

Watts, Allan W.

From: mpovman@bankofny.com
Sent: Thursday, January 26, 2006 1:33 PM
To: Watts, Allan W.
Subject: Re: Suporting Documents for Business Abstracts



SMDB Pricing API SMDB outbound SGA Systems smdb_interface_v6.SMDB tags.xls (71 MIDYS Translation
 Message Forma... message detail f... Arch.doc (307 KB) doc (316 KB)... KB) v10 - 08072...

----- Forwarded by Michael Povman/NY/DOMESTIC/BNY on 01/26/2006 03:32 PM

Michael Povman

To: Spencer Moser@BNY
 cc: Michael Povman@BNY,

Jim Reitz@BNY

03/02/2001 05:28

Subject: Re: Suporting
 Documents for Business Abstracts (Document link: Michael
 Povman)

Additional documents to support the BNY-JPM asset synchronization business abstract are attached.

Rgds,
 Spencer

(See attached file: SMDB Pricing API Message Formats.doc) (See attached
 file: SMDB outbound message detail for MIDYS v2.doc)

Spencer Moser
 03/02/2001 05:25 PM

To: Michael Povman@BNY
 cc: Jim Reitz@BNY (bcc: Spencer Moser)
 Subject: Suporting Documents for Business Abstracts

Attached are the functional and technical documents that support the BNY-JPM asset synchronization and SGA system busines abstracts. I expect to receive additional documentation from SGA. They have been experienceing production issues for two days, and have not had the opportunity to identify the appropriate documentation. We figured it was best to wait rather then send you everything, half of which may not apply.

Let me know if you have any questions.

Rgds,
Spencer

SGA

(See attached file: SGA Systems Arch.doc)

Asset Syncronization

(See attached file: smdb_interface_v6.doc) (See attached file: SMDB tags.xls) (See attached file: MIDYS Transalation v10 - 08072000.xls)

The information in this e-mail, and any attachment therein, is confidential and for use by the addressee only. If you are not the intended recipient, please return the e-mail to the sender and delete it from your computer. Although The Bank of New York attempts to sweep e-mail and attachments for viruses, it does not guarantee that either are virus-free and accepts no liability for any damage sustained as a result of viruses.

SMDB Pricing API Message Formats

This document describes message formats for SMDB Price and FX Rate feeds (implemented as Push Processors) provided to subscribing applications.

A message for an SMDB record represents a delimited alphanumeric string, that contains information about feed destinations and record fields. All message formats follow a tag/value sequence.

Tags are presented here with SMDB API declarations.

SMDB FX Rate Message Format

The following table describes SMDB FX rate tags

API DECLARATION	TAG	DESCRIPTION
AP_OBJ_SMDBFXRATE	"SMDFXRTE"	Opening Record Type Indicator
APPL_GSP	"^GSP^"	Feed to GSP
APPL_ASP	"^ASP^"	Feed to ASP
APPL_CDW	"^CDW^"	Feed to CDW
APPL_SGA	"^SGA^"	Feed to SGA
APPL_TAS	"^TAS^"	Feed to TAS
APPL_MAP	"^MAP^"	Feed to MAP
APPL_GSCS	"^GSCS^"	Feed to GSCS
APPL_JPM	"^JPM^"	Feed to JPM
-----	"^ROU^"	Tag Group Delimiter
API_TAG_FX_BEGIN	"^908^"	Opening Field Group Indicator
-----	"A"	"ADD" Action Flag
API_TAG_FX_CCY_FROM	"^909^"	"Converted From" Currency
API_TAG_FX_CCY_TO	"^910^"	"Converted To" Currency
API_TAG_FX_VND_ID	"^911^"	FX Rate Vendor ID
API_TAG_MARKET_ID	"^164^"	Market ID
API_TAG_EFF_DT	"^820^"	Effective Date
API_TAG_EFF_NY_TM	"^165^"	Effective New York Time
API_TAG_FX_TYPE	"^912^"	FX Rate Type (see below)
API_TAG_FX_BID	"^913^"	Bid FX Rate
API_TAG_FX_ASK	"^914^"	Ask FX Rate
API_TAG_FX_RULE_NO	"^915^"	Selection Rule No
API_TAG_RECORD_END	"^END^"	End Of Record Delimiter
API_TAG_EOM	"^EOM^"	End Of Message Delimiter

The order of the tags will be as presented in the above table, except that another tag group delimiter ("^ROU^") will be added at the very end (after "^EOM^"). The number of destinations will be in accordance with the SMDB FX Rate subscription list. FX Rate Type can have one of the following values:

VALUE	DESCRIPTION
“12”	12 Month FX
“9”	9 Month FX
“6”	6 Month FX
“3”	3 Month FX
“2”	2 Month FX
“1”	1 Month FX
“0”	Spot FX

The Selection Rule Number will be hard-coded to “1” pending further specifications. Other Action Flags will be specified as needed in the course of future development. A record example (for SGA) follows:

Example 1. FX Rate Record

```
SMDFXRTE^SGA^^ROU^^908^A^909^AED^910^ADP^911^0^164^1111
^820^1998-06-02^165^1998-06-02-13.55.06^912^0^913^0.1^914^0.1^915^1
^END^^EOM^^ROU^
```

SMDB Price Message Format

API DECLARATION	TAG	DESCRIPTION
AP_OBJ_SMDBPRICE	"SMDPRICE"	Opening Record Type Indicator
APPL_GSP	"^GSP^"	Feed to GSP
APPL_ASP	"^ASP^"	Feed to ASP
APPL_CDW	"^CDW^"	Feed to CDW
APPL_SGA	"^SGA^"	Feed to SGA
APPL_TAS	"^TAS^"	Feed to TAS
APPL_MAP	"^MAP^"	Feed to MAP
APPL_GSCS	"^GSCS^"	Feed to GSCS
APPL_JPM	"^JPM^"	Feed to JPM
	"^ROU^"	Tag Group Delimiter
API_TAG_REPEAT_SECID_BEGIN	"^791^"	Start of Numbering Schema/Purpose Code
API_TAG_SECID_BEGIN	"^793^"	Start of Security Type
API_TAG_SECID_COUNTRY	"^795^"	Country Code
API_TAG_FI_NBR_SYS_NO	"^818^"	Identification Number
API_TAG_NBR_SHM_TYP_CD	"^817^"	Type of Identification Number
API_TAG_NBR_SHM_PURP_CD	"^819^"	Purpose Code
API_TAG_EFF_DT	"^820^"	Effective Date
API_TAG_END_DT	"^821^"	End Date
API_TAG_RECORD_END	"^EOM^"	End of Security Type
API_TAG_REPEAT_SECID_END	"^792^"	End Of Numbering Schema/Purpose Code
API_TAG_SECID_END	"^794^"	End Of All Numbering Schemas
API_TAG_PRICE_BEGIN	"^860^"	Opening Field Group Indicator
	"A"	"ADD" Action Flag
API_TAG_INSM_NO	"^318^"	Instrument Number
API_TAG_EFF_DT	"^820^"	Effective Date
API_TAG_EFF_NY_TM	"^165^"	Effective New York Time
API_TAG_PRICE_VENDOR_ID	"^861^"	Vendor ID
API_TAG_MARKET_ID	"^164^"	Market ID
API_TAG_PRICE_PSR_NO	"^890^"	Price Selection Rule
API_TAG_PRICE_MKT_GRP	"^891^"	Market Group
API_TAG_PRC_CCY	"^834^"	Subscribed Currency
API_TAG_PRICE_TYPE	"^862^"	Price Type
API_TAG_PRICE_YIELD_FLAG	"^863^"	Price Yield Flag
API_TAG_PRICE_VALUE	"^864^"	Subscribed Price Value
API_TAG_MONTH_END_FLAG	"^865^"	Historical Price Flag
API_TAG_PRICE_ORIG_CCY	"^866^"	Original Currency
API_TAG_PRICE_ORIG_VALUE	"^867^"	Original Value
API_TAG_PRICE_EXCH_RATE	"^868^"	Price Exchange Rate
API_TAG_PRICE_TIME_STAMP	"^869^"	Time Stamp
API_TAG_RECORD_END	"^END^"	End Of Record Delimiter
API_TAG_EOM	"^EOM^"	End Of Message Delimiter

Example 2. Price Record

```
SMDPRICE^GSP^ROU^^791^^793^A^795^FR^818^FR0008771950^817^0^819^1^820^1989-08-
18^821^^END^^793^A^795^FR^818^4818287^817^2^819^1^820^1989-08-
18^821^^END^^793^A^795^FR^818^001073656^817^3^819^1^820^1989-08-
18^821^^END^^792^^791^^793^A^795^CH^818^XS0000609981^817^0^819^4^820^1998-06-
27^821^^END^^793^A^795^CH^818^0000609^817^2^819^4^820^1989-08-
18^821^^END^^793^A^795^CH^818^478131^817^4^819^4^820^1989-08-
18^821^^END^^792^^794^^860^A^318^^820^1998-09-
22^861^0^834^DEM^862^2^863^0^864^34.25^865^0^866^DEM^867^34.25^868^1^869^1998-09-22-
13.55.06^END^^EOM^^ROU^
```

A detailed explanation of the example follows:

SMDPRICE	Signifies that this is a pricing message.
^GSP	List of subscribing application identifiers, could be more than one.
^ROU^	Signifies a routable message
^791^	Signifies start of numbering schema message with same purpose code.
^793^A	Signifies start of a particular security type.
^795^FR	Country tag/value.
^818^ FR0008771950	Identification number tag/value.
^817^0	Type of Identification number tag/value. 0 signifies an ISIN.
^819^1	Purpose code tag/value.
^820^1989-08-18	Effective date tag/value.
^821^	End date tag/value.
^END^	End of particular security type message.
^793^...^END^	More security types.
^792^	End of numbering schema message with same purpose code.
^791^...^792^	More numbering schemas with same purpose code.
^794^	End of all numbering schemas.
^860^A	Signifies start of pricing message.
^318^75503	Instrument number tag/value.
^820^1989-08-18	Effective date tag/value.
^861^0	Vendor ID tag/value.
^834^DEM	Subscribed currency tag/value.
^862^2	Price type tag/value.
^863^0	Price yield flag tag/value.
^864^34.25	Subscribed price tag/value.
^865^0	Historical Price flag tag/value.
^866^DEM	Original currency tag/value.
^867^34.25	Original price tag/value.
^868^1	Exchange rate used tag/value.
^869^1998-09-22-13.55.06	Time stamp of message tag/value.
^END^	End of pricing message.
^EOM^	End of message.
^ROU^	End of route.

The SMDB outbound message (push record) has two distinct parts: The keys portion and the body portion.

Keys

- The Keys portion of the record is the beginning of the message. The end of the Keys portion of the message is represented by the tag, ^794^.
- Within the keys portion of the message, several distinct, logical repeating groups of data may appear. Each repeating group is started with the beginning tag ^791^ and ends with ^792^.
- Within the ^791^ and ^792^ beginning and ending tags, are the groups of logical data. Each group of data starts with the appropriate beginning tag and ends with the ^END^ ending tag. This is the data that requires translation. The tags ^791^, ^792^, and ^794^ do not require translation.

Body

- The body of the record contains the financial instrument record data, and begins after the ^794^ tag. The end of the body of the record is the ^END^ tag.
- Each piece of financial data has its own beginning tag, and is completed with the ending tag ^END^.
- The following is a sample list of beginning SMDB tags contained in the message:

Indicative data - SMDPUSH

- 793 - numbering schema begin tag
- 822 - financial instrument begin tag
- 823 - financial instrument characteristic begin tag. Example where financial characteristic is 20 = (^823^A^100^20^END).
- 824 - ratings begin tag
- 825 - depository begin tag
- 826 - related instrument begin tag
- 827 - financial component classification begin tag
- 849 - relationship begin tag

Corporate action data - SMDCORP

- 828 - corporate action schedule begin tag
- 843 - corporate action announcement information begin tag
- 838 - corporate action event number (char) begin tag
- 870 - corporate action election begin tag
- 880 - corporate action election component begin tag
- 900 - corporate action number (char) begin tag
- 950 - corporate action event component begin tag
- 975 - corporate action agent begin tag
- 980 - corporate action agent notification begin tag

SMDPRICE - price information

- 860 - market price information begin tag
- 793 - numbering schema begin tag

Notes

- There will be 1 value (usually an 'A') following the beginning tag.
- All tags are surrounded by carrots '^' (i.e., ^793^)
- Sequence of groups within a portion of the message does not matter, as each group is independent of the next.
- '^EOM^' denotes the end of the message, and will appear after the final ^END^ tag.
- Each tag, except 791, 792 and 794, END and EOM will have values, delimited by carrots. However, if no value exists, this is not an error. Simply, the value does not exist for this record.

Example: “^821^^END^” – In this example, the 821 tag does not have a value and is followed immediately by ^END^ tag.

- For the complete list if the tags, refer to the updated MIDYS Translation document.

Example of Message from SMDB to MIDYS

```
SMDSELCTGSP.....^791^^793^A^795^CH^818^000023896^817^4^819^
1^820^1980-01-01^821^^END^^793^A^795^CH^818^023896^817^4^819^1^820^1980-01-
01^821^^END^^793^A^795^GB^818^0024464^817^2^819^1^820^1980-01-
01^821^^END^^793^A^795^GB^818^820011997^817^1^819^1^820^1997-12-
05^821^^END^^793^A^795^GB^818^GB0000244649^817^0^819^1^820^1997-12-
05^821^^END^^792^^794^^822^A^318^869^336^0^G01^0003^G1E^1HARP(ALBERT E.) AES
EUROPEAN INCOM^G11^SHARP(ALBERT E.) AES EUROPEAN
UNIT^G12^TRUST^402^UNKNOWN^673^10064^135^98^404^GB^G16^0^601^28^403^GBP^651^GB
P^444^0.00010^316^0.00010^125^43^812^GB^804^1.00000^END^^823^A^100^167^END^^823^A^100
^230^END^^825^A^G30^91^820^1997-12-
05^G32^Y^END^^827^A^318^869^H01^0^H02^0^H03^62^H04^15262^END^^827^A^318^869^H01^0
^H02^0^H03^1000^H04^0^END^^EOM^
```

Example of Message from SMDB to MIDYS Logically Grouped for MIDYS Reference

Note: There is no line feeder in the actual message.

```
SMDSELCT
GSP
.....
^791^          <----- BYPASS
^793^A
^795^CH
^818^000023896
^817^4
^819^1
^820^1980-01-01
^821^
^END^
^793^A
^795^CH
^818^023896
^817^4
^819^1
^820^1980-01-01
^821^
^END^
^793^A
^795^GB
^818^0024464
^817^2
^819^1
^820^1980-01-01
^821^
^END^
^793^A
^795^GB
^818^820011997
^817^1
^819^1
```

^820^1997-12-05
^821^
^END^
^793^A
^795^GB
^818^GB0000244649
^817^0
^819^1
^820^1997-12-05
^821^
^END^
^792^<----- BYPASS
^794^<----- BYPASS
^822^A
^318^869
^336^0
^G01^0003
^G1E^1HARP(ALBERT E.) AES EUROPEAN INCOM
^G11^SHARP(ALBERT E.) AES EUROPEAN UNIT
^G12^TRUST
^402^UNKNOWN
^673^10064
^135^98
^404^GB
^G16^0
^601^28
^403^GBP
^651^GBP
^444^0.00010
^316^0.00010
^125^43
^812^GB
^804^1.00000
^END^
^823^A
^100^167
^END^
^823^A
^100^230
^END^
^825^A
^G30^91
^820^1997-12-05
^G32^Y
^END^
^827^A
^318^869
^H01^0
^H02^0
^H03^62
^H04^15262
^END^
^827^A
^318^869
^H01^0
^H02^0

^H03^1000
^H04^0
^END^
^EOM^

Message Details

- In the first byte of the message, you will find the function name (SMDSELECT) which must be translated into IMNTACK in the SWIFT 598 header message type field.
- Starting from position 9 to 12, the destination, JPM must be present.
- Positions 13 through 84 can be bypassed.
- Starting from position 85, you will find the tag-delimited SMDB push record, as indicated above.

Header

On the message from MIDYS to JPM, SMDB requires that MIDYS create SWIFT-like header with the following information:

```
{4:  
:20:123456  
:12:598  
:77E:/MSG_TYP/IMNTAK  
/DATA_SRC/SMDB  
/DATA_TRGT/LITE  
/PBMN/123456  
/POST_DAT/19990709  
/POST_TIM/16:00:00EST
```

The Bank of New York

Strategic Global Accounting

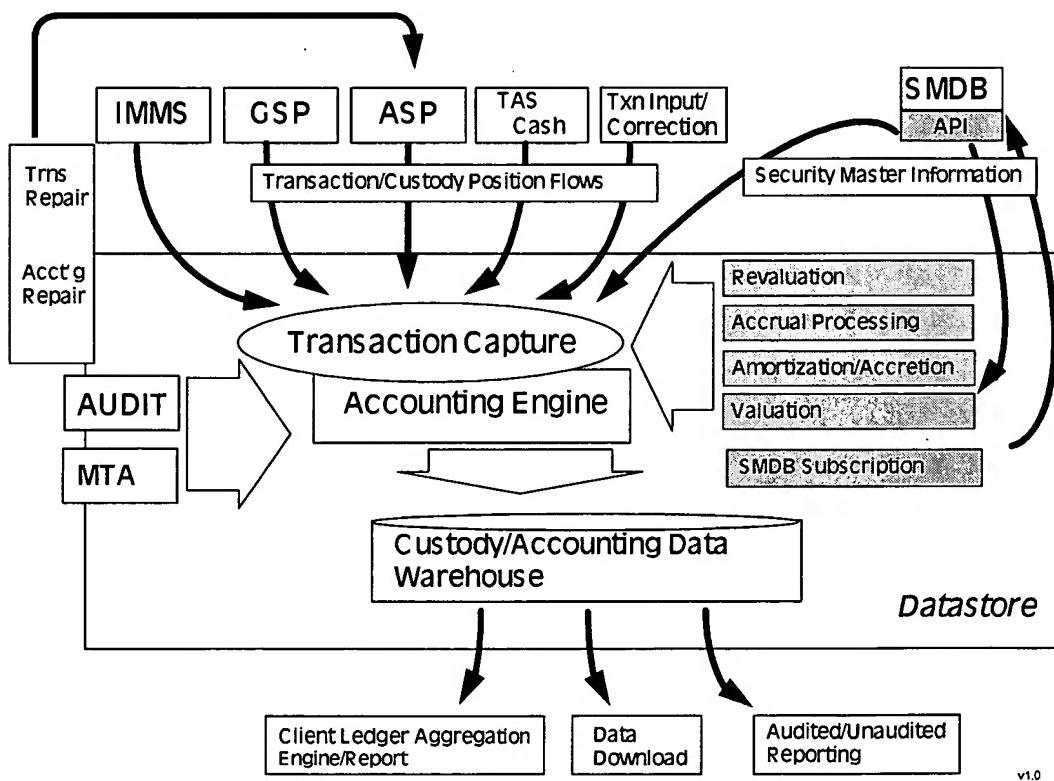
System Architecture

Introduction

The Strategic Global Accounting system will be an online, real time system, providing full double entry, general ledger accounting. The core accounting engine will run under CICS, with the database housed in DB2 on the mainframe. The system will provide full 24 hour per day functionality for all markets and time zones.

Turning the Datastore into an accounting system

In order to leverage the vast amount of data and processing that currently exists in the Datastore, it seemed appropriate to build the accounting functionality within the Datastore architecture.



The means of turning the Datastore into an accounting system essentially involve replacing the batch feeds of accounting information that come from the two accounting systems, MAP and TAS, with real time processing based on the feeds of transactions from ASP, GSP, IMMS, TPFX and TAS CASH. The transaction feeds would be passed to a new real time accounting engine which would create the value added accounting information on the transaction, and post that transaction to positions within the database.

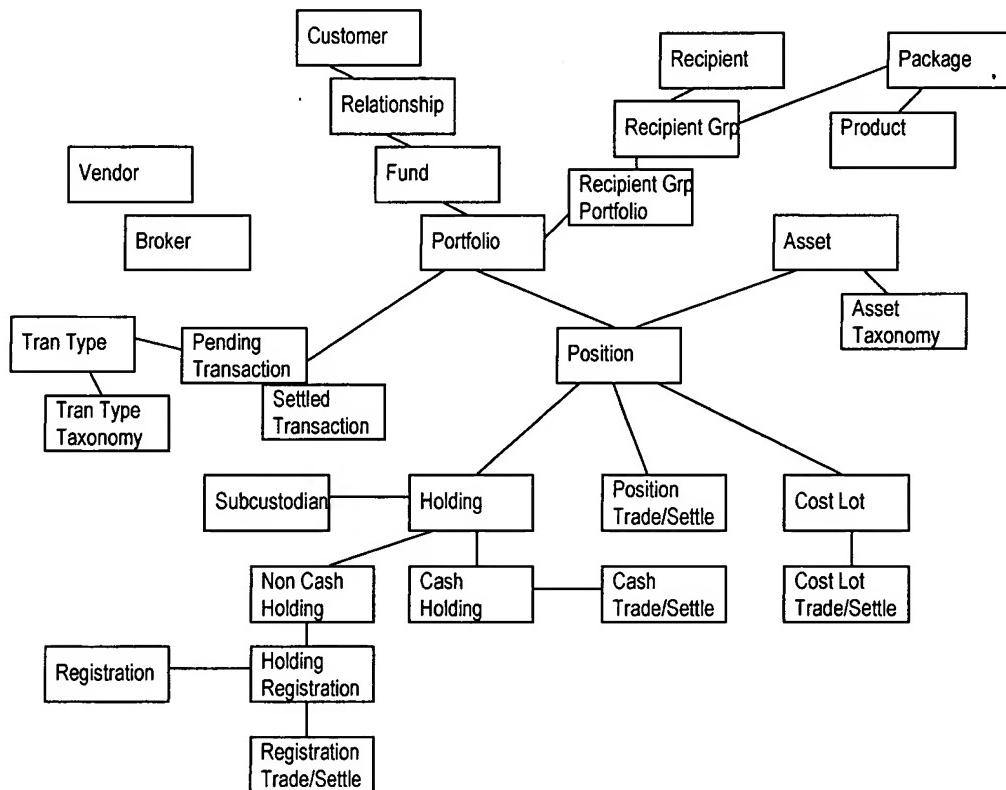
The split of responsibilities will be as follows. The capture of transactions will be handled as phase one of the effort that will ultimately provide a Custody Data

Warehouse. The transaction will be captured and, if the portfolio requires accounting, it will be passed to the Global Accounting system.

There is some amount of preprocessing that will take place in CDW. Primarily, the Custody Data Warehouse system will convert transaction input from the upstream format into Datastore format. The transaction will be typed using a standard format, and relevant reference data, such as Asset, will be maintained.

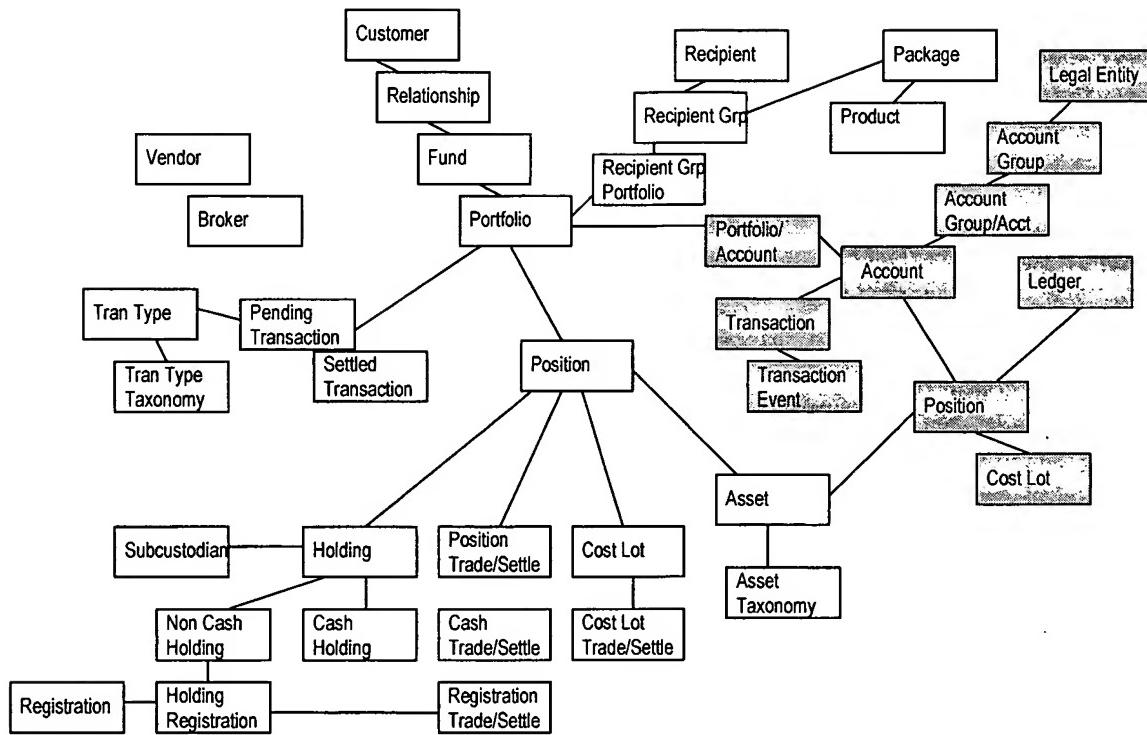
The Datastore is made of over 200 tables of various data types. Four different Data Types exist, **Demographic** (ie., Customer, Relationship, Portfolio, Recipient), **Product Delivery** (ie., Package, Product, Schedule), **Reference** (ie., Asset, Vendor, Broker), **Financial** (ie., Position, Transaction, Holding). Each of the four areas are represented in this simple depiction of the Datastore database.

Datastore Current Environment



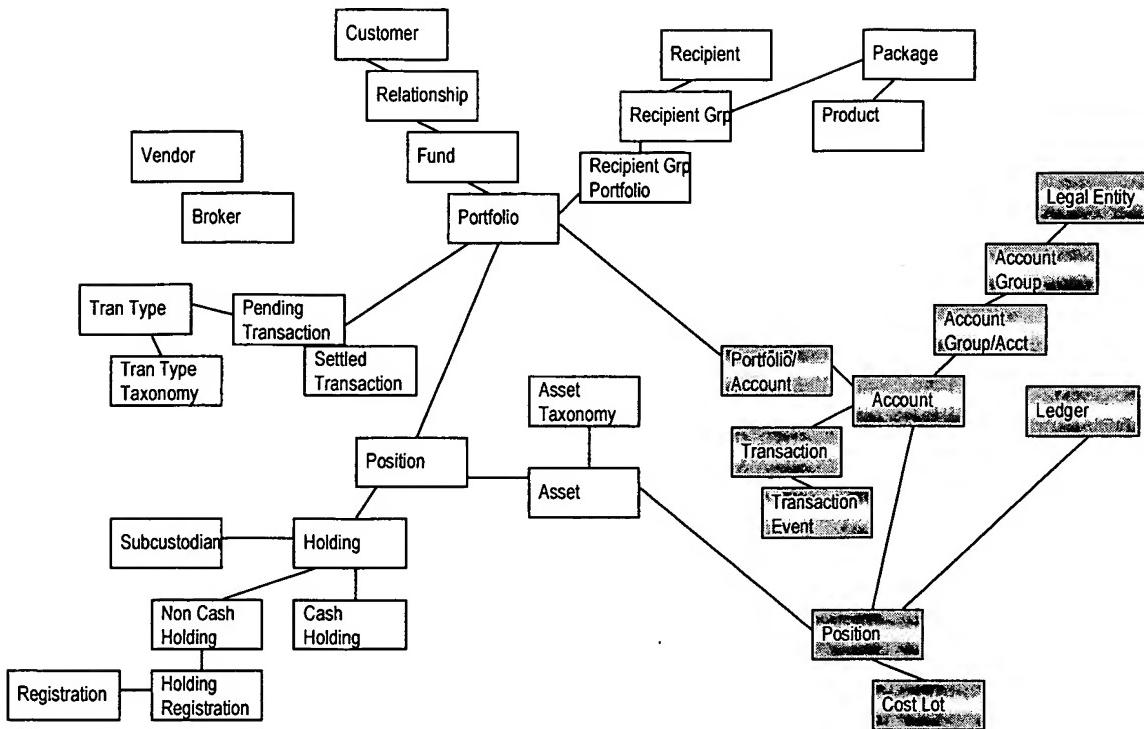
The addition of Global Accounting adds a series of new tables into the environment, largely focusing around Transaction and Position.

Datastore with SGA Integration



Ultimately, as Global Accounting functionality replaces that of MAP and TAS, the Position and Transaction related entities that are populated via the current accounting interface to Datastore will be dropped.

Datastore after MAP/TAS Feed Shutdown



What is left is a Warehouse of securities data. The Financial related data entities will be split in two. Custody data will be captured and housed in the "Holding" leg of the database, being fed directly from the Bank's SMAC systems. Accounting related data will be populated through functionality developed and maintained within Global Accounting. There will be single sources for Reference, Demographic and Product Delivery data which will be shared by both systems.

Double Entry General Ledger Accounting

One of the first things that you would learn in an Accounting 101 course are the basic tenets of Double Sided General Ledger Accounting, that Assets = Liabilities + Owners Equity. The Assets, Liabilities and Owners Equity are broken down into a series of subsets called Accounts. These are not Accounts as we know them here in the Bank, but rather accounting ledgers. To avoid confusion, we more often refer to them as ledgers. Asset ledgers are normally considered Debit accounts, meaning that their balances are increased through a Debit entry. Liability and Owners Equity ledgers are considered Credit accounts. A positive balance in a Debit account is considered a Debit Balance as a positive balance in a Credit account is considered a Credit Balance. Therefore, the sum of all the Debits and Credits in the portfolio must equal zero. Traditional accounting methodology holds that for each transaction there are a series of Journal Entries made. At least one of these entries must be a Debit and one a Credit and the sum of the Debits must equal the Sum of the Credits.

Historically here in the Bank of NY we have done Single Sided Accounting. Both Map and Tas have inventory and detail files respectively that represent the number of a

particular financial instrument held and how much it cost. It is as if the Asset side of the Asset = Liability + O.E. were being accounted. Where there is liability accounting done, it is most likely to be handled as a negative asset. There is usually no treatment for Owners Equity.

Unlike the two current accounting systems of the Bank of New York, SGA will provide for full double entry, general ledger accounting. The way that general ledger balances will be achieved is actually quite simple. Currently, an accounting position is essentially the intersection between an account and an asset. For instance, account 123456 has purchased some Bank of New York stock. This coming together of an account and an asset results in a position. In fact, general ledger accounting simply expands the intersection that results in position from two entities, account and asset, to three -- account, asset and ledger. Now, account 123456 has a position in Bank of New York stock in the Investments Held ledger.

A position that may have looked like this...

Position

Account	Security	Units	Cost
123456	Bank of New York Common Stock	1000	36150

...now becomes much more descriptive when the Ledger Identifier is added. A glance will tell you which aspect of this trade date position is settled and which is still receivable.

Position

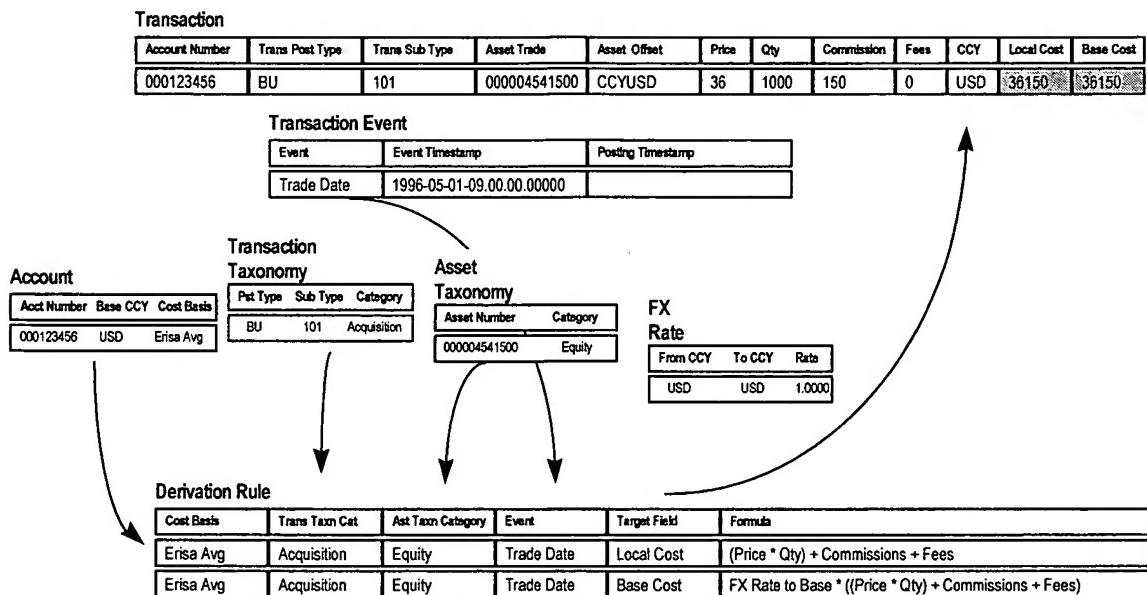
Account	Security	Ledger	Units	Cost
123456	Bank of New York Common Stock	Investments at Cost	500	17000
123456	Bank of New York Common Stock	Receivable	500	19150

The Accounting Engine

The two most elementary aspects of accounting, determination of cost and update of position, are represented within the Accounting system by two separate processes, Derivation and Posting.

Derivation is the act that essentially takes the basic business transaction, (ie., A buy of 1000 shares of Bank of New York Stock at \$36/share plus \$150 in commissions and fees), and adds the accounting specific information, such as Local Cost, Base Cost, and Gain/Loss.

Derivation of Common Stock Buy on Trade Date



The rules for derivation are table driven. Each rule provides the formula that is used to determine an output amount which will be placed in the target field of the transaction record. The input into the formula can come from a number of places. Primarily, the input is provided from the stub of a transaction that is passed to Global Accounting from Custody Data Warehouse. But input is also provided from the Account table, the Transaction and Asset Taxonomy tables, the FX Rate table, and, as is the case for a disposition, from the Position table.

The set of rules that are executed depend on a number of factors. First, what Cost Basis is the Account using? Here we are showing one, Erisa Average Cost, but the system will be able to handle up to four concurrently. The set of rules that are executed depend on a number of factors. Also, what type of transaction is being executed? To minimize the number of rules that need to be maintained, the posting for similar types of transactions is grouped by means of a Transaction Taxonomy. In this case, the buy transaction with the Post Type of BU and a Sub Type of 101, is categorized as an Acquisition.

The Asset or Financial Instrument that is traded also plays a role in the derivation of the transaction. In the same way that a category of transactions are grouped, a category of assets are grouped via an asset taxonomy. Here, the asset, 000004541500, which is the identifier for Bank of New York Stock, is classified in the posting taxonomy as an Equity.

Finally, the set of derivation rules varies depending upon which point has been reached along the transaction life cycle. Here, the Event which has been reached is trade date, the first event that will be posted for the transaction. For an Erisa Average Cost Basis on Trade Date an Acquisition of an Equity will execute the two included formulae to determine the Local Cost and Base Cost.

Once the transaction has gone through the Derivation step, it is ready for Posting, which takes the result from the Derivation process and updates Ledger balances.

Posting of a Common Stock Buy on Trade Date

Transaction

Account Number	Trans Post Type	Trans Sub Type	Asset Trade	Asset Offset	Price	Qty	Commission	Fees	CCY	Local Cost	Base Cost
000123456	BU	101	000004541500	CCYUSD	36	1000	150	0	USD	36150	36150

Transaction Event

Event	Event Timestamp	Posting Timestamp
Trade Date	1996-05-01-09.00.00.00000	1996-05-01-09.00.15.00000

Quantity	Local Cost	Base Cost	Local Cost

Posting Matrix

Cost Basis	Trans Taxn Cat	Event	Ledger	Effected Asset	DR	CR
Erisa Avg	Acquisition	Trade Date	Units	Asset Trade	DR	
Erisa Avg	Acquisition	Trade Date	Inventory at Cost, Local	Asset Trade		DR
Erisa Avg	Acquisition	Trade Date	Inventory at Cost, Base	Asset Trade		DR
Erisa Avg	Acquisition	Trade Date	Payable for Securities, Local	Asset Offset		CR
Erisa Avg	Acquisition	Trade Date	Payable for Securities, Base	Asset Offset		CR

Inventory at Cost, Local 000004541500 -- BK	Inventory at Cost, Base 000004541500 -- BK	Payable Securities, Local CCYUSD	Payable Securities, Base CCYUSD	Units 000004541500 -- BK
5/1 36150	5/1 36150		5/1 36150	5/1 1000

The rules for Posting a transaction, which are embedded in the "Posting Matrix", are similar in their application to the Derivation rules. Like the Derivation rules, the Posting rules are keyed by Cost Basis, Transaction Taxonomy Category and Event. Any Asset related differences in posting will be handled in the derivation process, removing the need for asset related Posting rules.

For each combination of Cost Basis, Transaction Category and Event there may be zero or more rules that would be triggered. Here, five different Posting rules are initiated for the Bank of New York purchase. Each rule specifies the Ledger balance that will be affected. That balance will be either Debited or Credited with the amount in the transaction field identified across the top right hand side of the Matrix. Since the Ledger balances are kept at the most granular level, meaning there is a set of ledgers for each position held, the Asset or Financial Instrument that should be posted needs also to be specified. The rule itself tells us to post to either the Asset Trade or the Asset Offset. The Asset Trade in this case is Bank of New York Stock. The offset is United States Dollars. (Note: Normally the offset is some type of currency. However, there is the possibility for a Barter trade of some sort that must be handled systematically.)

In the Posting exhibit you will see the list of rules, the appropriate Debit or Credit indicator for each field within the transaction, and below that there appears the graphic

representation of the ledgers that have been effected, and their Debit or Credit balance
(Note: Debits on the left, Credits on the right.)

Three days later, when the purchase settles, a new event is posted to the transaction. The Settlement Date event has its own set of posting rules that need to be applied. First, derivation would occur for the transaction based on a set of Settlement Date rules. In this simple single currency example there are no further derivation functions that need to take place, so the physical transaction would not be updated. In a true multicurrency example there would likely be an FX Gain or Loss on the cash amount involved in the transaction based on exchange rate fluctuations during the three days. The important consideration, however, is that any single field on the transaction is populated by only one single derivation process. Once the field is populated, it cannot change, information can only be added not modified.

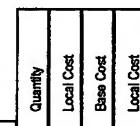
Posting of a Common Stock Buy on Settlement Date

Transaction

Account Number	Trans Post Type	Trans Sub Type	Asset Trade	Asset Offset	Price	Qty	Commission	Fees	CCY	Local Cost	Base Cost
000123456	BU	101	000004541500	CCYUSD	36	1000	150	0	USD	36150	36150

Transaction Event

Event	Event Timestamp	Posting Timestamp
Trade Date	1996-05-01-09.00.00.00000	1996-05-01-09.00.15.00000
Settlement Date	1996-05-04-05.00.00.00000	1996-05-04-05.00.15.00000



Posting Matrix

Cost Basis	Trans Taxn Cat	Event	Ledger	Effected Asset	Quantity	Local Cost	Base Cost	Local Cost
Erisa Avg	Acquisition	Settle Date	Units	Asset Offset	CR			
Erisa Avg	Acquisition	Settle Date	Inventory at Cost, Local	Asset Offset		CR		
Erisa Avg	Acquisition	Settle Date	Inventory at Cost, Base	Asset Offset			CR	
Erisa Avg	Acquisition	Settle Date	Payable for Securities, Local	Asset Offset		DR		
Erisa Avg	Acquisition	Settle Date	Payable for Securities, Base	Asset Offset			DR	

Inventory at Cost, Local CCYUSD	Inventory at Cost, Base CCYUSD	Payble Securities, Local CCYUSD	Payble Securities, Base CCYUSD	Units CCYUSD
5/4 36150	5/4 36150	5/4 36150 5/1 36150	5/4 36150 5/1 36150	5/4 36150

Posting of the transaction on Settlement Date will occur following the Derivation process. In this example you can see how the amounts that had been credited to the two payable ledgers have now been offset by two equal debit entries. In addition, posting entries have been made to the Inventory at Cost ledgers as well as the Units ledger for US Dollars.

While the ledger methodology provides for tremendous flexibility in the number of and types of balances that can be stored and accounted for within the system, the vast majority of inquiries to accounting data will be looking towards a subset of ledgers that

are common across all security types. Items like Units, Cost and Accruals are likely to be demanded together. To facilitate this requirement and speed performance certain logical ledgers will be physically stored together.

Units 000004541500 – BK	Inventory at Cost, Local 000004541500 – BK	Inventory at Cost, Base 000004541500 – BK	Accrued Income, Local 000004541500 – BK	Accrued Income, Base 000004541500 – BK	Broker Commission 000004541500 – BK
1000	36150	36150	500	500	1500

Ledger

Ledger	Ledger Type	Parent Ledger	Position Field
Units	Debit	Trade Date Position	Units
Inventory at Cost, Local	Debit	Trade Date Position	Local Cost
Inventory at Cost, Base	Debit	Trade Date Position	Base Cost
Accrued Income, Local	Debit	Trade Date Position	Accrual Base
Accrued Income, Base	Debit	Trade Date Position	Accrual Local
Broker Commission	Credit		

Position

Account Number	Asset	Ledger	Balance	Units	Local Cost	Base Cost	Accrual Base	Accrual Local
000123456	BNY Stock	Trade Date Position	0	1000	36150	36150	500	500
000123456	BNY Stock	Broker Commission	1500	0	0	0	0	0

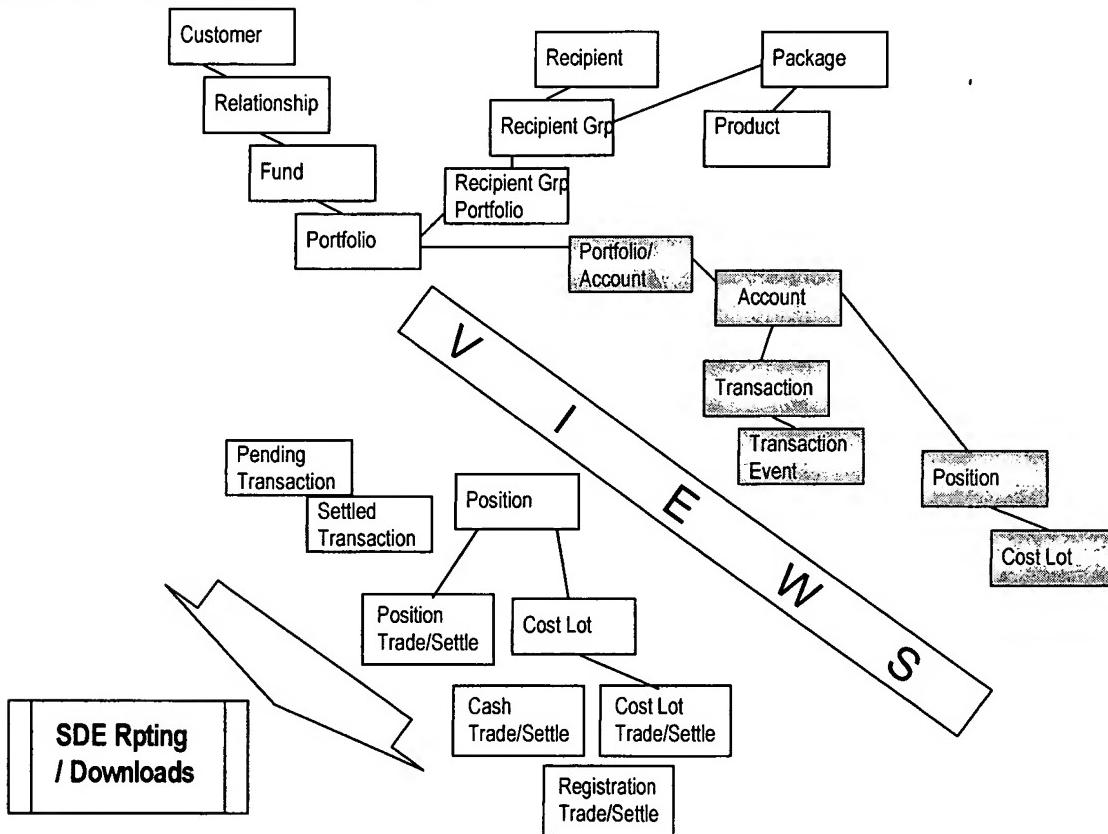
The six ledger balances that are symbolically represented above will actually appear in the database as two rows in the position table. One row, contains values in five different fields: Units, Local Cost, Base Cost, Accrual Base and Accrual Local. Each of these fields is representative of a ledger balance. The Ledger table is the road map used to determine where the logical ledger value is physically stored. For instance, the row in the ledger table for "Inventory at Cost, Local" shows that it is a Debit ledger and that it has a Parent Ledger called "Trade Date Position". That means that in order to find the value for "Inventory at Cost, Local" you need to look in a position row that has the key equal to "Trade Date Position" – the parent ledger. There are a number of fields on the position table. The one that we need to find is identified within the ledger row for "Inventory at Cost, Local" in the Position Field column. Here we see that "Local Cost" is the position field with which we are interested. Therefore, to find the ledger balance for "Inventory at Cost, Local" we go to the position row with the key, "Trade Date Position" and look for the value in the field "Local Cost". It's just that simple.

When the ledger is both logical and physical, as is the case for Broker Commission, there is no parent ledger nor position field populated for that row on the ledger table. Instead, the ledger value is found on a position row with the partial key of ledger id equal to the logical ledger in a field called Balance.

One of the key requirements of the system will be the ability to integrate the data that results from the accounting process with the existing library of processes that exist under the Datastore banner. The two key areas that must be addressed are transaction and position. A key database design goal has been to keep as much of the new data designs closely aligned with the existing datastore.

By having standard "position" rows available as physical items within the database, it enables the system to be more compatible with the existing Datastore product delivery mechanisms. The "Trade Date Position" row that will exist in SGA is not all that different from the position row that currently exists within the Datastore. A series of views will be created within SGA that enable the report and download programs that currently run within the Datastore to read the SGA data as well.

Datastore Reporting of SGA Data



While some of the detail that is available with double entry general ledger would be lost within the single sided Datastore reports and downloads, they do provide a production ready vehicle for providing customer information and performing reconciliation.

The rules for Deriving and Posting transactions will be established for each line of business that the Bank services. Timestamping of the rules will allow for their change without the necessity of a conversion. The rule sets will be completely "viewable" by the accounting staff, minimizing the potential for confusion. And since the accounting rules are table driven as opposed to being embedded in a series of programs, they may be

used within the Audit process itself, eliminating one possibility for error -- that the same program that posted a transaction is not the one that spots an out of proof condition.

One reason to separate Posting and Derivation is to facilitate "As Of" processing. The matter of finding a position for any point in time becomes a very mechanical exercise, where each transaction/event combination that has occurred between any desired points in time is "unposted".

24x7 Functionality

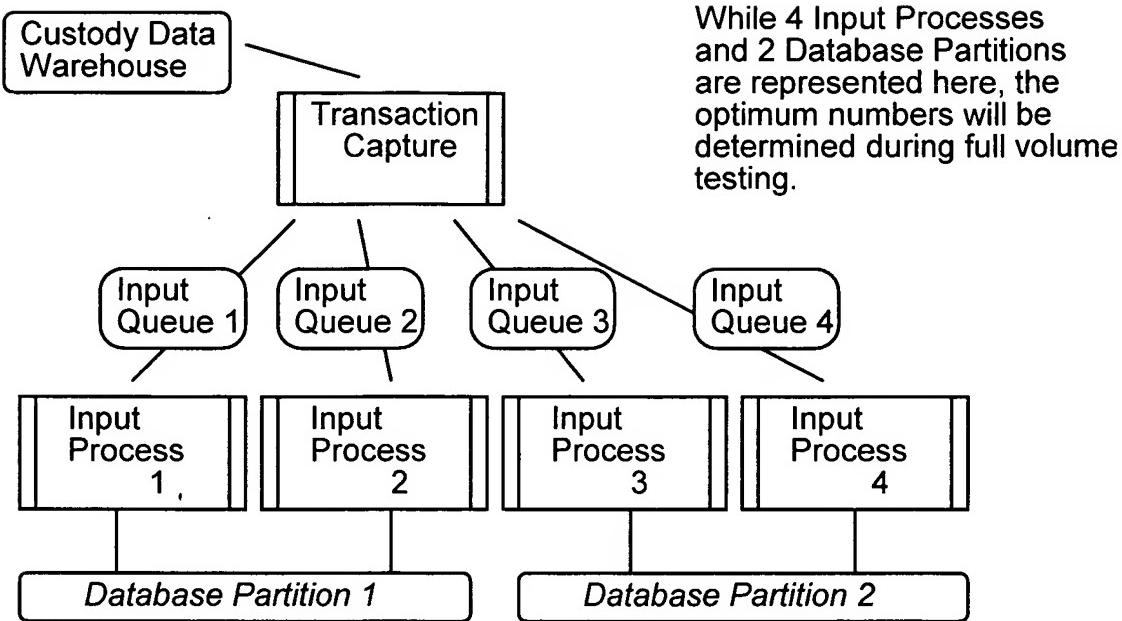
The Global Accounting system will be operational 24 hours a day, 6½ days a week. Reorganization of the database will take place during a short down time which will most likely occur on Sunday morning.

Scalability

The core requirement of the system will be 24 hour per day real time accounting. The question is, how do you achieve this requirement, particularly when there are stated goals of 1,000,000 transactions per day with a peak of 100,000 in an hour.

While it is in no way feasible at this point in time, consider how fast the accounting systems could run if we split the account base they supported in half and ran dual systems concurrently with each half of the accounts. In fact, such a method would work from a business standpoint, as long as only one process were operating against an account at any point of time. The sequence that transactions post is very important, but only within the context of a single account. Therefore, the theoretical limit to the number of concurrent accounting systems you could run would be equal to the number of accounts you had in the bank. While running that number of processes would be impossible, imagine running 4, or 8 or 40 concurrent versions of TAS, each with a handful of accounts. You would think that the resultant savings in over night batch time would be fantastic. The problem is that with the underlying database manager that supports both TAS and MAP, we would very quickly reach a level of diminishing returns as the concurrent executions of the accounting systems competed for resources. In fact, the batch time would most likely be increased, not decreased.

DB2 is much more capable than VSAM for handling concurrent operations. Global Accounting will meet the goals of 1,000,000 transactions per day and 100,000 an hour, by the simultaneous processing of multiple input capture procedures. In addition to multiple input processes, the database itself will also be partitioned, minimizing the potential for delay. At least one CICS region, dedicated to SGA, will be devoted to the high volume throughput of transactions.



Broken down by ranges of account numbers, these input procedures will receive transactions from two primary sources. The first of these sources is Custody Data Warehouse, which will capture transactions from the Bank's Cash Management and Security Movement and Control systems.

The second source of transactions will be the Accounting system itself. The only way an accounting balance may be updated is by way of a posted transaction. Therefore, scheduled processes within Global Accounting, such as Accruals and Amortization, will create adjustment transactions to affect the proper balances. There are a number of benefits that will be derived from this methodology. First, the updating of positions will be limited to the posting process that is encapsulated within the input procedure. Second, there will be a complete audit trail for every balance updated within the system. Third, the triggering and completion of these scheduled processes can take place concurrent with the real time posting of financial transactions, without disrupting the 24x7 nature of the system.

Flexibility

Flexibility is what will make The Bank of New York's Global Accounting System superior to the competition. The system has been designed so that in each key area there is the flexibility that is necessary for the Bank to quickly respond to the needs of a changing marketplace.

Each of the following contingencies can be quickly addressed:

New Line of Business

Each set of accounting rules will be defined within the bounds of a Line of Business. As a new business line is identified, its differences from other established lines will be identified. The likelihood is that the combination of one or more lines that will already be defined will cover the vast majority of rules associated with the new business. Different

aspects of these existing lines will be used as models from which the new set of new rules will be established. It is important to note, that no lines of programming code would need to change to support the new line of business.

New Transaction Type

All Derivation and Posting rules will be defined at the Transaction Category level. Once a basic classification is established, there would likely be a category which would conform to all Derivation and Posting rules required. Given that, the systems effort that would be required to add the new tran type to the existing category is essentially nil. Should the new transaction show some characteristics that are somewhat different than existing categories, then a new category can be created. The new transaction category would need to have new Derivation and Posting rules assigned to it, and this could probably be accomplished to a large extent by modeling it after one or more existing categories. In either case, the majority of time required to add the new category is in business analysis -- no programming changes would be necessary.

New Asset Type

The development of the Security Master Database will play a large role in the development of the Global Accounting System. One of the key features of the SMDB system will be the new method of Asset Classification. No longer will the accountant or other business personnel be dependent on the current Taxonomy system that is largely driven by obscure codes and programming that is essentially hidden. In the near future all Security Classification will be completely rules driven and to a large extent, automated. As new securities are added to the database, their placement in a category will be determined by an easily accessible set of rules. The accounting rules that contain a security dimension will be associated to a category. As new securities are introduced to the system, they will be placed in one of the existing categories, inheriting this category's set of posting rules. Should a financial instrument be established that has characteristics which are completely different than those of any established set of securities, a new category would be established via the tools provided by SMDB. As with the Transaction Category addition, the accounting rules for this new Asset Category would be created largely from the existing set of rules that will already exist. Once the simple modifications are established for this new instrument, preparation is complete and the system will be ready to do full accounting.

New Chart of Accounts

There will be one internal Bank of New York Chart of Accounts used for posting and auditing. However, the Ledger balances that are kept are at a very granular level. In fact, each balance is maintained on a security by security basis. Since our balances will be maintained at a level more detailed than external clients would require, we can "roll up" or aggregate these ledgers in form that the client would need to see. The Aggregation Engine, as it is called, will map the internal Chart of Accounts to the clients desired Chart. Not only is this a benefit from the reporting standpoint, but it will also serve as a very valuable audit tool, enabling the accountant to view the client's Chart as well as the Bank's Chart during the normal review process.

Processing Overview

Input Process

The Strategic Global Accounting (SGA) system will process and store the accounting related transaction and event information produced by the upstream Trading Systems (ASP, GSP, IMMS, TPFX, etc.) and passed by them into the Custody Data Warehouse (CDW). Transaction and event data will be read from the Custody Data Warehouse, formatted, and passed to SGA in real-time. Input transactions will also be brought in from internal scheduled accounting processes and SGA on-line screens.

The input cycle will have various edit checks for transaction/event data content and validation. Any transactions which fail the edits will be held in a repair queue for correction.

The input process will also determine the correct sequencing of transactions as they are read. Out of sequence transactions have the capability of dynamically triggering the reconstruction process, depending on system and customer parameters.

Derivation and Posting

As transactions are handled by the input process, certain events will cause dynamic derivation of accounting data such as lot takedown, various cost basis values and gain/loss. Customer parameters will affect which sets of rules and data are to be used for specific groups of transaction types and financial instruments.

Transactions will dynamically be posted to the appropriate ledger positions as they go through the normal event cycle. The posting rules are specified according to basic accounting principles and uniquely identified by groups of transaction types, transaction event, type of financial instrument, and customer parameters.

Scheduled and Related Processes

The Initialization Process will run once a day and will refresh a control Queue with the schedules for the current day.

Scheduled Process Initiator will poll the "Current Day" Schedules, comparing them to the current System Time/Date. It will insert a row into the Portfolio Scheduled Table marking the Current Day Schedules as "Initiated".

There will be a number of Preprocessors Polling the Portfolio Scheduled Table, one for each Task (e.g. Accruals, Amortization, Valuations etc.). The Preprocess will assemble a Partially Augmented transaction for each Account and Position combination and will insert it into a Swept Accounts Queue. The entries in the Queue will lack SMDB data. While the Preprocessor is running, it will lock the Account, preventing the Input Process from posting a transaction that can affect the Account. SMDB data will be filled in by the

specialized Process that will in effect complete the transaction and put the results onto MQ Series Queue where it will be picked up by Input Process. The Process will interface with SMDB using the pre-defined set of APIs.

Accounting Functions

Global accounting requirements are identified by customer/account group/account parameters and general accounting rules for transaction types. The processes required for global accounting are identification and calculation of accounting values pertinent to the input transaction (derivation) and the application of these values to the appropriate ledger positions based on a matrix of posting rules.

The process of transaction derivation is done only once for any transaction, although the timing of this process may be dependent on customer-related and transaction-related conditions (e.g. trade-date vs. settlement-date accounting). For customers with lot-based accounting, the affected position lots for the transaction are identified at the time of derivation.

The posting rules are specified according to basic accounting principles and uniquely identified by groups of transaction types, transaction event, type of financial instrument, and customer parameters. A matrix of these rules will identify which ledgers will be affected by any transaction event.

The ledger position updates are always synchronized with transaction event processing in order to maintain correct balances in the chart of accounts. A timestamp will be retained with each posted transaction event to allow journalization of positions.

Transaction Flow

The Custody Data Warehouse will be the repository of SMAC transactions. An SGA program, running as a background task, will periodically sweep the appropriate CDW database tables for transactions and/or events which have not yet been transmitted to SGA. These will be formatted for use by SGA, written to the outbound message queue (MQ Series) to be routed to SGA, and then marked as having been sent. Each of the transactions will be routed to a different logical queue based on an account number range. This is to allow concurrent processing within Global Accounting.

On the Global Accounting side, a separate background task will be running to process the messages retrieved from each of the logical queues. Although separate tasks are running, each performs the same functions using the same set of programs. The separation is done strictly for processing performance.

Each transaction or event received will be processed within one logical unit of work (L.U.W.). Processing will include editing, and as necessary, matching, derivation, reconstruction, posting. At the end of the unit of work, the transaction will be syncpointed to ensure database integrity.

Transactions or events which fail the edits, or for other reasons require operator intervention will be routed to an internal database table (identified as the **repair queue** in the system architecture diagram) for subsequent repair processing.

As a part of the accounting process, a real-time link to the Security Master Database (SMDB) will be supported.

Performance and Optimization

The SGA system is designed for high volume, real-time processing. A number of features in the system architecture are to be implemented to achieve the anticipated transaction volume of up to 100,000 input transactions per hour. In order to allow SGA to achieve this volume level, a separate CICS region, with connections to CDW and SMDB, will be established. Should this prove insufficient, the system architecture allows for the establishment of multiple CICS regions, with an identical configuration, running the same programs.

Transactions which are "swept" from the Custody Data Warehouse will be routed to one of many logical outbound queues for processing by SGA. Each queue will process transactions and events for a limited account number range. The exact number of logical queues will depend upon available system resources, but fifty to one hundred is likely, more may be possible.

On the SGA side, each logical queue will be associated with one background task exclusively dedicated to processing, in sequence, transactions which arrive on its queue. This allows for concurrent processing of different transactions, separated by account number range. This precludes the possibility of contention for a given portfolio by two or more input processes.

A locking strategy, described in more detail elsewhere in this document, will prevent portfolio updating conflicts between the input process and/or various scheduled background processes.

Input Sources

The primary input data source for Global Accounting is the **input queue** from the Custody Data Warehouse. This is a real-time input source, the volume and arrival timing of which is unpredictable. There are other sources of data, generated within the SGA system itself.

Scheduled background processes will route account/position data to the SMDB system for amortization, accrual, and valuation. After SMDB processing, these will be routed back into the same **input queue** as that used by CDW data, and will be processed in the same manner. This is potentially the highest volume input source. This is a time-driven process.

Transactions and events which were routed to the **repair queue**, when repaired and release by an operator, will be fed back into the front end of SGA, as if they were coming from external sources.

Transactions which are internally generated as a part of a reconstruction process are routed to another internal queue (identified as the **reconstruction queue** on the system architecture diagram). The reconstruction queue has a higher priority than that of other input sources and will be processed immediately after creation.

The third internal queue (identified as the **hold queue** on the system architecture diagram) is used to store input transactions which could not be processed when received because the portfolio involved was being used by another process. They also will be fed back into the front end of SGA.

Input Prioritization

Since we have a number of different input sources, from both external and internal sources, a prioritization strategy is required. The SGA online input module will handle this prioritization.

The highest priority input source will be the reconstruction queue. This is a dynamic queue created when a regular input transaction or event is recognized as triggering a reconstruction situation. The events and transactions it contains will be read and processed sequentially until the queue is drained, at which time the queue will be deleted. This is the highest priority because the portfolio position is now out of sequence, and should be reordered before any further posting is done.

Initially, the second priority will be the repair queue, since it is assumed an operator is waiting for the results of the repair. The repair queue process has not been fully defined at this time. It is very possible that the repair queue itself may have multiple priorities, depending on what is being repaired.

The third priority is the hold queue. Every transaction is to be processed in the sequence it is received (repairs excepted). Transactions on the hold queue, which were already received, but could not be processed at that time, will take precedence over any new transactions received.

The fourth priority will be incoming CDW transactions and the scheduled transactions received from SMDB. The exact order of these two sources will be determined later.

If because of this prioritization a transaction arrives out of sequence, it will be recognized as an "as-of" transaction, generate reconstruction, and thus be processed correctly.

Throughout this discussion of priorities, it should be remembered that we will have multiple concurrent tasks in process, delimited by account range. Although one portfolio may be undergoing reconstruction, it will not impact the processing of other accounts outside of the account range.

It is acknowledged that this prioritization is subject to revision upon further analysis.

Reconstruction

If a transaction arrives that is recognized as "as-of" (chronologically out of sequence), the transaction/event processor module will invoke the **reconstruction** process to rebuild a portfolio's position in the correct sequence.

The reconstruction process consists of generating reversal transactions into a dynamic, recoverable queue, which will bring a position to the point just prior to the "as-of" transaction's correct point in time. It will sequence the "as-of" transaction for derivation and posting, and regenerate the transactions which were reversed. These regenerated transactions will be rederived prior to posting.

All of these transactions will be written to the **reconstruction queue**, for subsequent processing in order. At this point, the portfolio will be locked for reconstruction and a checkpoint taken to preserve data integrity. Processing invoked by the "as-of" transaction will terminate. The input control module will then initiate the processing of the reconstruction queue, in its entirety, before accepting any other incoming transaction.

Locking Strategy

The SGA environment will be designed to support a volume of up to 100,000 transactions per hour, while insuring data integrity. Since more than one process and type of process will be running concurrently, a locking strategy is required.

The input process of reading separate queues by account range, will ensure that the same portfolio (account) will not be processed by two concurrent tasks. There remains the issue of other background processes (such as scheduled accrual, amortization, valuation, snapshot) which work with portfolios independent of what is being done by the input programs. To ensure data integrity, a new table will be implemented to control update access to a portfolio.

The locking strategy will be defined for each process in the system. It is recognized that while some processes (such as accrual and amortization) can run concurrently, other processes (such as input) require exclusive control of the portfolio. Each process will lock any portfolio it is to process in this table. Other non-concurrent processes will not attempt to work with a locked portfolio. Instead, they will requeue that portfolio to be processed when it is no longer locked by another task.

If the portfolio is not available, the input transaction will be written to the internal **hold queue** for later processing.

On-line Recovery

The standard CICS and DB2 recovery capabilities will be used to ensure overall database integrity and to preclude the loss of any transaction or event. The input process, running as a single unit of work, will syncpoint at the end of its work, if successful. If not successful, the transaction and any database updates will be rolled back, and the starting transaction requeued for subsequent processing.

The Data Collector

The Collector Process designed to facilitate the collection of data that exists within the GA universe. It is a mechanism to assemble data for virtually any process and allows the requester to be completely independent from the Physical architecture of SGA.

Regular mode is a one for one scenario when one request results in one set of information returned (Per Asset). Batch mode on the other hand may return a result set of one or more layouts (Multiple Assets). Both modes return information for one Account.

The primary purpose of the Collector is to provide an independent means of providing ledger balance information. It will also be able to collect SMDB data and perform a required business calculations. That means that the Derivation process for example will get all the data it needs to do the cost derivation, but the math will be done in Derivation process itself. Eventually the user will have a choice to request Math to be perform by Collector by way of a stored procedure.

Control Monitoring

Recognizing that the majority of the functionality within Global Accounting is provided by background CICS tasks, an on-line inquiry capability will be provided to allow authorized users to monitor the status and processing statistical information captured by those tasks. CICS screens will display system status, queue record counts, as well as current portfolio processing states.

Program Description

Input Processing

CDW Sweeper	This program will read intraday transactions from the Custody Data Warehouse, format them for SGA, and write them to the outbound message queue to be routed to SGA.
SGA Control	This module will be the driver module for SGA transaction processing. It will call the appropriate sub-programs as required. It is an on-going background task, working with a specific logical message queue. Its termination is the end of a single unit of work. Each called program will indicate, via return codes, the success or failure of the called process. The control program will also update the various control information statistics for user display.
Input	This will read transactions from either the CDW input stream (via MQ Series) or from one of the several internal database queues according to priority.
Editing	This program will edit the input transaction to ensure that all data present in the transaction is valid. Input records which fail the edit will be written to the repair queue (a DB2 table) for subsequent user correction.
Repair	This function will allow users to correct or add additional required information (repair) a transaction which passed CDW edits, but failed those of SGA. When repaired, such transactions will be released for reprocessing. When analysis of the repair function is further along, it is likely that this will prove to be multiple programs.
Matching	This program will verify that any linked transactions required to process this input are already present within the SGA environment.

Program Description

Input Processing (continued)

Transaction or Event Processor	This program will analyze an input transaction/event and determine the processes it needs to be correctly posted. These processes may include reconstruction, derivation, etc. It will call the appropriate sub-programs as required. These sub-programs will include dynamic amortization and accrual capabilities.
Reconstruction	This program will generate the appropriate transaction reversals and transactions required for reconstruction. They will be written to the internal reconstruction queue for subsequent high-priority processing for this account. A return code passed back to the control program will indicate that this unit of work should end. No further processing of the original input transaction (which caused reconstruction) will be done at this time. The control program will then again call the portfolio control module to have it lock this portfolio for reconstruction.

Program Description

Accounting

- | | |
|---------------|---|
| Lot Selection | This program will identify specific position lots associated with an accounting transaction if the customer preferences indicate lot-based accounting requirements. The program will create transaction lot keys related to the transaction for each identified position lot. |
| Derivation | This program will augment a transaction with accounting data required for ledger posting. The derivation processes to be applied to a transaction will be determined by rules based on transaction type, type of financial instrument, and customer preferences. |
| Posting | This program will update ledger balances by applying a fully-augmented accounting transaction to a pre-defined posting matrix which identifies the ledger balance to be affected. |

Program Description

Scheduled Processing

Initialization	Initialization is needed to provide the full set of schedules for the Current Day. Once the Schedules are identified, they will be Swept onto the Current Day Queue, where the Initiator will be able to pick them up. This Will limit Initiator's Search for Required Schedules and will allow for monitoring of Daily Events. After completing the Sweep, Scheduler will mark the account as Processed. When all of the Accounts have been "Processed" the Initialization Module will clear the Queue and will Re-populate it with the Next Day Schedules.
Initiator	This program will run in a background restarting itself every N seconds. N will have a default value and will be periodically reset internally by the Initiator. It will select the certain time entries off of the Current Day Schedules Queue and will Lock the accounts by inserting the entries into Accounts Scheduled Queue. Once accounts for the given schedule time are Selected, the Initiator will mark the entries in Current Day Schedules Queue as Complete. The N variable can be reset to the time interval shorter then the default if needed. One process will handle every potential schedule to reduce the number of potential locks against the Accounts Scheduled Queue.
Preprocess	Preprocess is a Specialized Function that performs only one task. Every Functionality will have a separate Sweeping Process associated with it. The Preprocess will Poll the Accounts Scheduled Q, Lock Accounts, and will construct the Complete Transaction. The Preprocess will rely on Collector Process to Accumulate the Account/Position/Ledger values it needs and on a Universal Control Module to regulate its access to the Account's data. The Preprocess will Unlock the Account when all the positions within it are processed. Preprocess will be designed to run as a background task or it may be linked to. Preprocess will Know if the Account to Sweep is Locked or Not by calling a Universal Control Module. If it is Locked the account will remain "not swept".
Process	This Module Completes the Transaction by calling accrual, amortization, valuation or other module. Like Preprocess this module is specialized in performing only one business task. Process has no knowledge about the location of the Data required to complete the transaction. That knowledge belongs to a Specialized Accrual, Amortization or other API provided by SMDB.
Dynamic	The Set of Dynamically Called Modules will be developed to satisfy every possible request from Input, Derivation, Posting or Online System.

Program Description

System Utility

Portfolio Control	This program will be called to ensure that no other process will work with a given portfolio while the input transaction is "in-flight", a background task is in progress, or some other activity requiring exclusive control is going on.
System Monitor	This program will display the system's transaction statistics and current state via an on-line screen.

THIS PAGE BLANK (USPTO)

THIS PAGE BLANK (USPTO)

Error! Not a valid link.

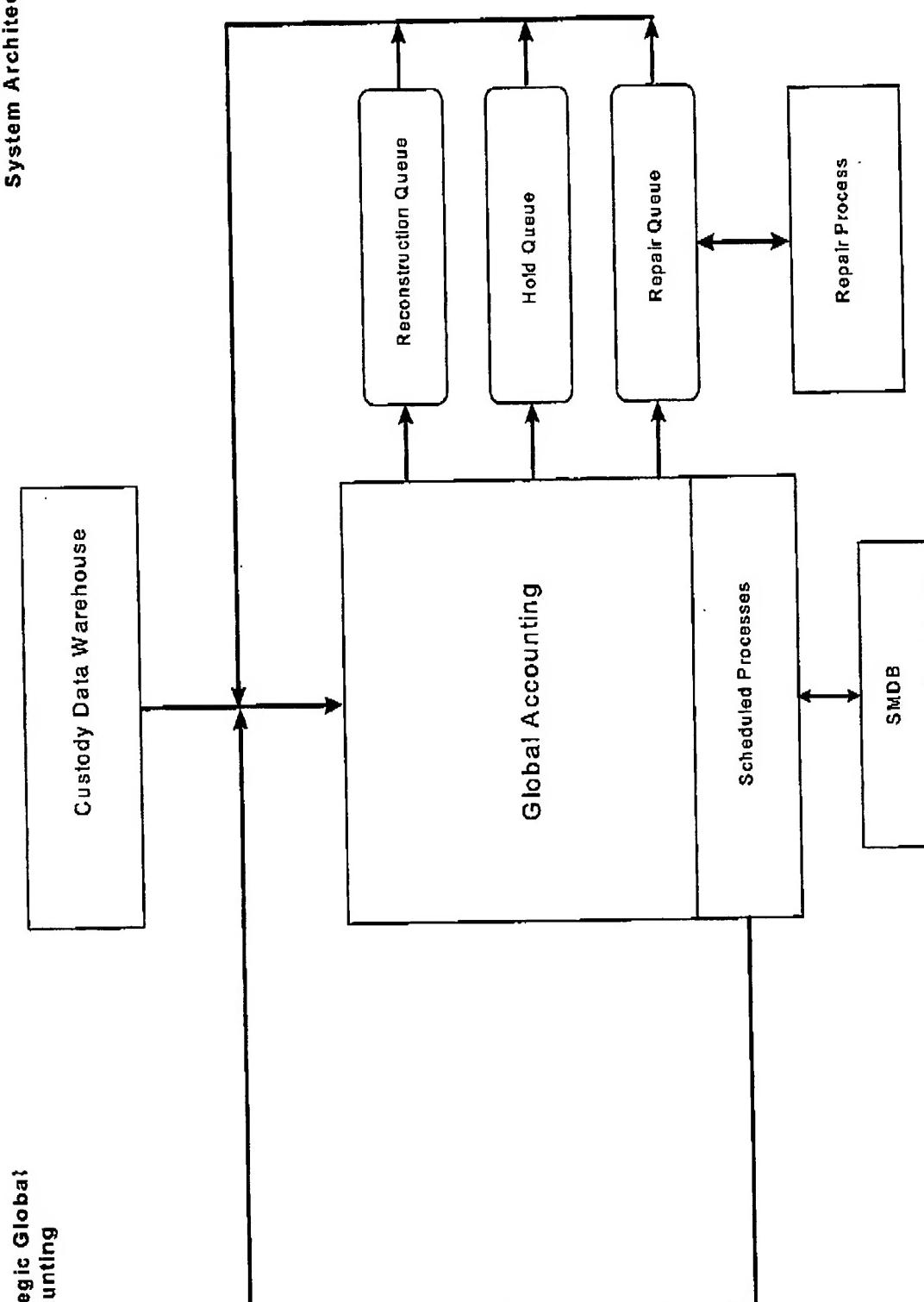
THIS PAGE BLANK (USPTO)

Error! Not a valid link.

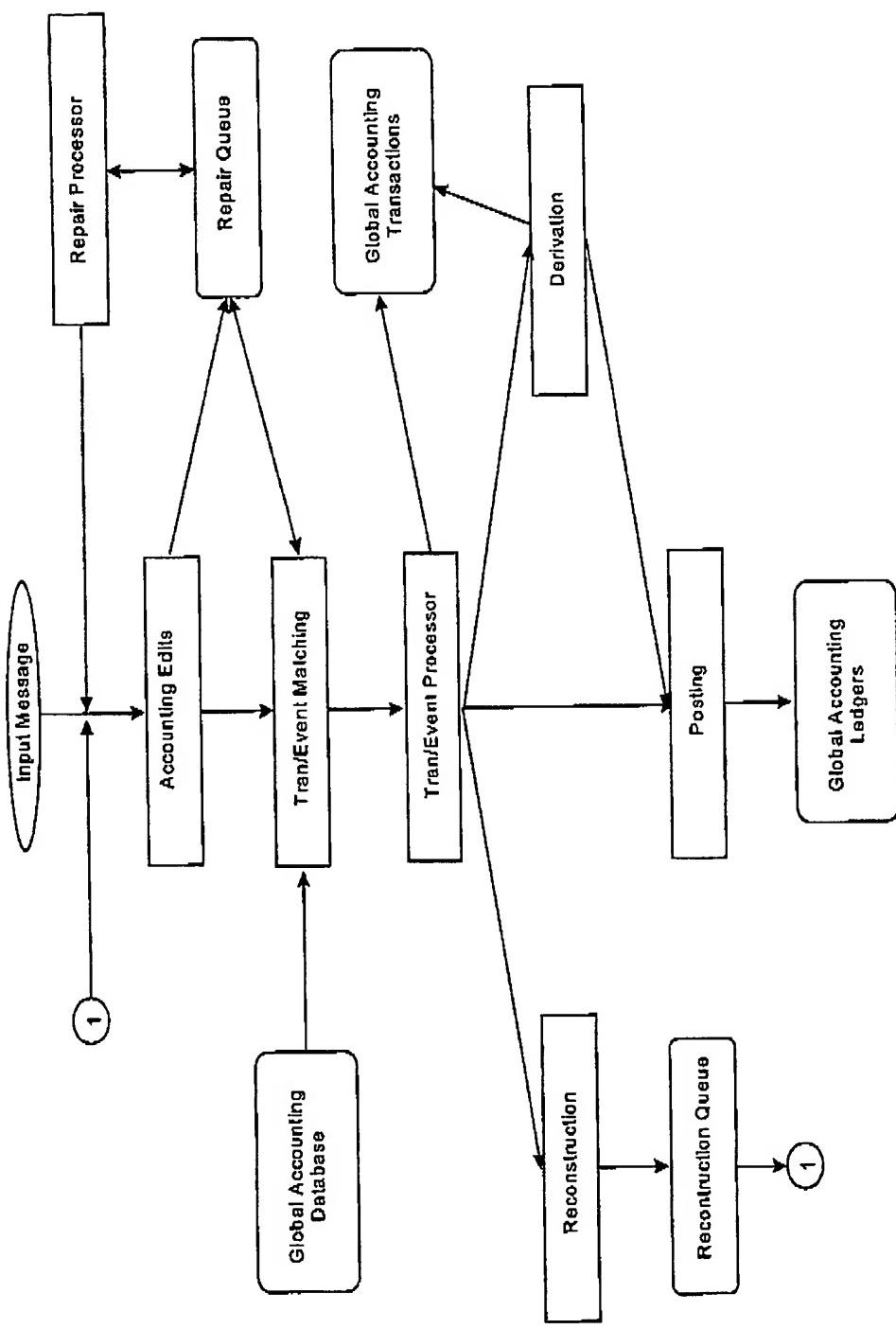
THIS PAGE BLANK (USPTO)

Error! Not a valid link.

EV478774021US

System Architecture

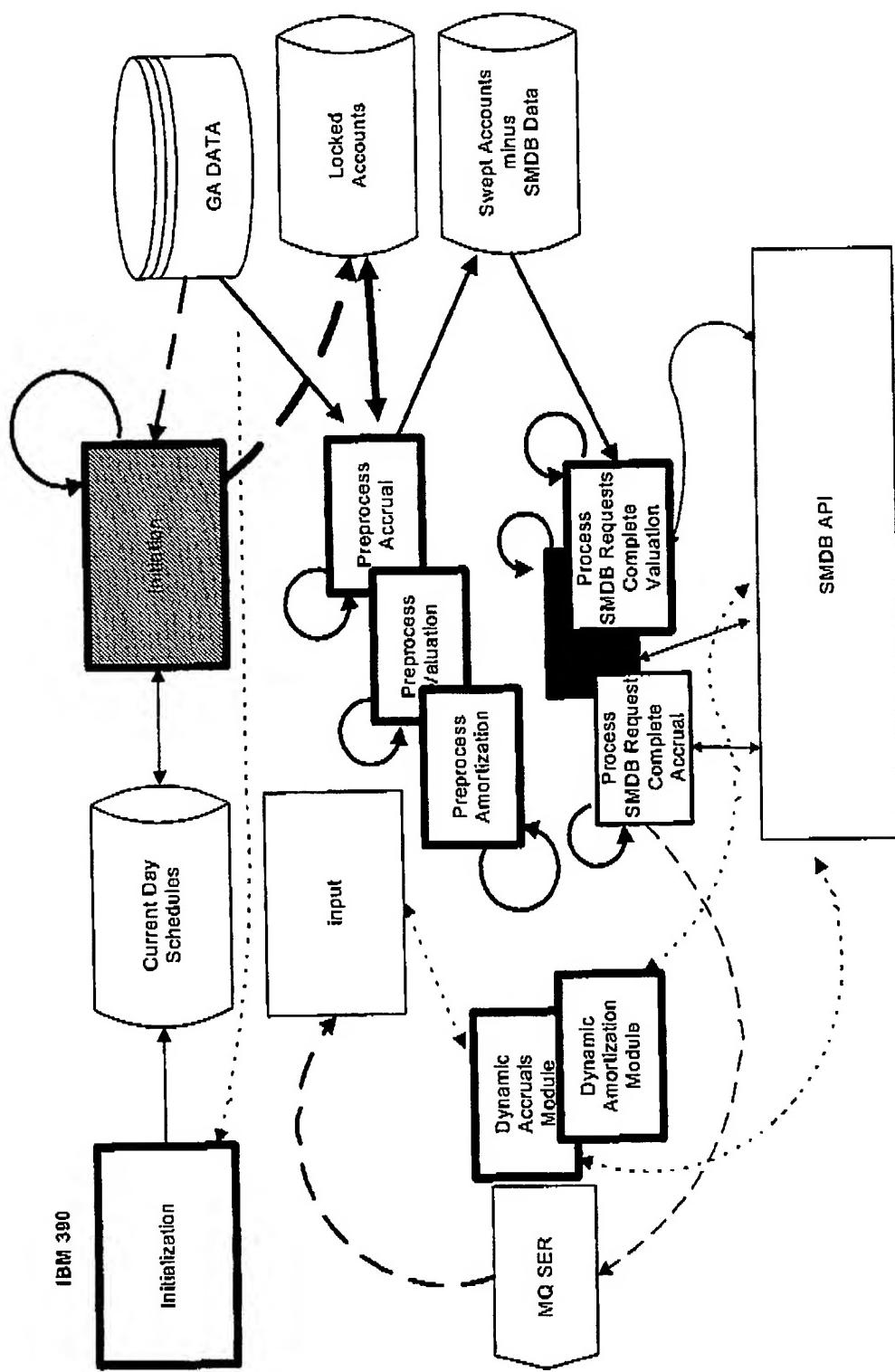
EV478774021US

Strategic Global Accounting**Input Transaction Flow**

EV478774021US

Strategic Global Accounting

Scheduled Processes



EV478774021US

Why

Accounting Requirements

Securities Processing Architecture

Turning the datastore into an accounting system

The Datastore is made of over 200 tables of various data types. The data can be split up into three different types, Demographic (ie., Customer, Relationship, Portfolio, Recipient), Product Delivery (ie., Package, Product, Schedule), Reference (ie., Asset, Vendor, Broker), Financial (ie., Position, Transaction). In order to leverage the vast amount of data and processing that currently exists in the Datastore, it seemed appropriate to build the accounting functionality within the Datastore architecture.

The means of turning the Datastore into an accounting system. Essentially involve replacing the batch feeds of accounting information that comes from the two accounting systems, Map and Tas, with real time processing based on the feeds of transactions from ASP, GSP, IMMS, TPFX and TAS CASH. The transaction feeds would be passed to a new real time accounting engine which would create the value added accounting information on the transaction, and post that transaction to positions within the database.

The split of responsibilities will be as follows. The capture of transactions will be handled as phase one of the effort that will ultimately provide a Custody Data Warehouse. The transaction will be preprocessed and if the portfolio requires accounting it will be passed to the Global Accounting system.

The preprocessing in CDW? Convert from upstream format into Datastore format. Transaction typing. Maintenance of relevant reference data.

Integrating new data to existing datastore functionality

Datadownloads

Reporting

Taxonomies Asset and Transaction

Data Flows

everything happens with a transaction

transaction flows

CDW role

Input Processing

Reconstruction

Internally Generated transactions

reconciliation

Processing Architecture

Database Architecture

Keys to Position -- General Ledger

The Chart of Accounts

Derivation Rules

Posting Rules

Multiple Base Currencies

Multiple Cost Bases

Aggregation of Ledgers

Implementation

benefits of using the Datastore

compatability

Transaction data is very similiar to that of the Datastore. Most information is contained on a single table entry.

If that is not possible, there will be the ability built into the system, to dynamically create

ASSET SETUP AND MAINTENANCE

INTERFACE SPECIFICATION

Approved by: Amal Chaudhuri.....

Ahmed Sharif

Accepted by: Safi Bajwa

Wayne Giles

Naguib Khoury

Crhis Letta

Version 1.01	June 3, 1999	Naguib Khoury
Version 1.02	June 16, 1999	Spencer Moser
Version 1.03	June 18, 1999	Naguib Khoury , Spencer Moser
Version 1.04	July 1, 1999	Naguib Khoury , Spencer Moser

ASSET SETUP AND MAINTENANCE INTERFACE SPECIFICATION.....	3
MESSAGE TYPES.....	3
<i>Phase 1 Message Types.....</i>	<i>3</i>
<i>Phase 2 Message Types.....</i>	<i>4</i>
<i>Phase 3 Message Types.....</i>	<i>4</i>
MESSAGE FORMATS	5
COMMON FIELDS	5
IMNTADD	5
<i>Design.....</i>	<i>5</i>
<i>Sample Instrument Add Tags for Equity, Warrants, Rights.....</i>	<i>6</i>
<i>Sample Instrument Add Tags for Corporate, Government.....</i>	<i>7</i>
<i>Sample Instrument Add Tags for Mortgages.....</i>	<i>8</i>
<i>Sample Instrument Add Tags for Municipalities</i>	<i>10</i>
<i>Sample Instrument Add Tags for Private Placements</i>	<i>11</i>
<i>Sample Instrument Add Tags for Exchange Traded Derivatives.....</i>	<i>11</i>
IMNTMOD.....	12
IMNTSUB	13
IMNTREC	13
IMNTACK.....	14
DATA TRANSLATION AND INCLUSION IN IMNTACK.....	14
IMNTNAK	15
IMNTLST	15
IMNTPRICEPUSH AND IMNTCAPUSH	16
SAMPLE MESSAGES.....	17
IMNTADD EQUITY.....	17
IMNTADD GOVT	17
IMNTADD MBS	18
IMNTADD MUNI	19

Asset Setup and Maintenance Interface Specification

Message Types

SWIFT-like message formats will be used to convey asset setup, acknowledgement and maintenance information between AMS and BNY. In order to process a message conforming to basic rules set forth by the major flows for the Liberty project, SWIFT MT598 message type will be used.

Where possible, SWIFT rules¹ will be used with user-defined tag-words provided in this document. ISO standards will be used for all currencies, countries, SWIFT standards will be used for values and dates.

Message types will be created and tested for all asset setup functionality in phases, as indicated below.

Phase 1 Message Types

AMS initiated messages

Tag Value	Description	Tag	Tag Word
IMNTADD	Instrument Add Message Type	77E	MSG_TYP
IMNTMOD	Instrument Modify Message Type	77E	MSG_TYP
IMNTREC	Instrument Full Record Sync Message Type	77E	MSG_TYP

Note: IMNTREC will be tested in phases. In phase 1, IMNTREC has the same functionality as IMNTADD and reconciliation will consist of AMS sending their universe of London-held securities to SMDB, approximately 7,000 securities. BNY will accept the IMNTREC and perform reconciliation based on this universe. Breaks will be handled on the break-queue and an exception report will be produced.

Subsequent phases associated with much larger reconciliation volumes will require a more elegant method of transmission and requires further discussion.

BNY initiated messages

Tag Value	Description	Tag	Tag Word
IMNTACK	Instrument Acknowledge Message Type	77E	MSG_TYP
IMNTNAK	Instrument Negative Acknowledge Message Type	77E	MSG_TYP

Notes:

- IMNTACK is the full SMDB push record that acts as a confirmation of an AMS initiated IMNTADD or IMNTMOD.
- IMNTNAK is initiated by MIDYS, the BNY middleware provider, not SMDB. MIDYS responds to an AMS IMNTADD on a technical level in the case of receiving an incomplete or unrecognized IMNTADD.

¹ As defined by SWIFT International Standard ISO 15022 Data Field Dictionary

Phase 2 Message Types

AMS initiated messages			
Tag Value	Description	Tag	Tag Word
IMNTSUB (for pricing data)	Instrument Subscribe Message Type	77E	MSG_TYP
IMNTREC	Instrument Full Record Sync Message Type	77E	MSG_TYP

BNY initiated messages			
Tag Value	Description	Tag	Tag Word
IMNTPRICEPUSH	Price Data Push Message Type	77E	MSG_TYP

Phase 3 Message Types

AMS initiated messages			
Tag Value	Description	Tag	Tag Word
IMNTSUB (for corporate action data)	Instrument Subscribe Message Type	77E	MSG_TYP
IMNTLST	Instrument Reference Data Update Message Type	77E	MSG_TYP

BNY initiated messages			
Tag Value	Description	Tag	Tag Word
IMNTLST	Instrument Reference Data Update Message Type	77E	MSG_TYP
IMNTCAPUSH	Corporate Action Push Message Type	77E	MSG_TYP

Note: All dates will be provided in the forthcoming Liberty project plan.

Message Formats

Common Fields

A SWIFT MT598 message will be used, and various fields will be common across all asset messages. Where possible, SWIFT rules will be adhered to in the common areas of the message. AMS will send a unique technical number from the messaging layer to BNY on each message, and AMS will receive a unique technical number on all BNY messages transmitted to AMS. The following fields are common across all IMNT messages and act as a message header with control information:

Note: A SWIFT-like delimited format is used for the tag words and data portion of the message. Forward slashes will be used to delimit tag words from their corresponding data values followed by <ctl><lnd>, as follows:

Status	Tag	Liberty Tag	Description	AMS / ILITE TAG
M	20		Transaction Ref Number: Message layer unique technical number	
M	12		Message description tag MT598	MSG_ID
M	77E		<CTRL><LNFD>	
M	77E	MSG_TYP	Instrument Message Type	MSG_TYP
M	77E	DATA_SRC	Data Source Code	DATA_SRC
M	77E	DATA_TRGT	Data Target	DATA_TRGT
M	77E	PCMN	AMS Technical Number	PCMN
M	77E	POST_DAT	Posting Date	POST_DAT
M	77E	POST_TIM	Posting Time	POST_TIM
M	77E		<CTRL><LNFD>	

IMNTADD

Design

The IMNTADD message is transmitted from AMS to BNY for asset add requests. The message will be in SWIFT-like MT598 format, embedding a single tag/value token within the field 77E narrative. Each instrument add request will correspond to a single SWIFT message. Tag values will correspond to mnemonics agreed upon between AMS and BNY. Each tag word and data value set will end with a <ctl><lnd> until the end of the message.

The following tables define the contents of the IMNTADD message based on SMDB field requirements for public securities. There are six major breakdowns as detailed in the attribute spreadsheet compiled during the AMS / BNY working sessions (The attribute spreadsheet can be found under separate cover). The message is variable length, as certain tags apply only to certain asset classes.

BNY will respond to IMNTADD messages with an IMNTACK message containing the attributes detailed in the IMNTACK section of this document. AMS will then reconcile the reflected attributes with the attributes originally transmitted to BNY.

The six types of IMNTADD requests are:

- Instrument add for Equity, Warrants, Rights
- Instrument add for Corporate, Government

- Instrument add for Mortgages
- Instrument add for Municipal
- Instrument add for Private Placements
- Instrument add for Exchange Traded Derivatives

Sample Instrument Add Tags for Equity, Warrants, Rights

IMNTADD for Equity, Warrants, Rights (public securities)

Status	Tag	SWIFT FIELD	Description	AMS / LITE TAG
M	20		Transaction Ref Number: Message layer unique technical number	
M	12		Message description tag MT598	MSGID
M	77E		<CTRL><LNFD>	
M	77E	MSGTYP	Instrument Message Type	MSGTYP
M	77E	DATASRC	Data Source Code	DATASRC
M	77E	DATATRG	Data Target	DATATRG
M	77E	PCMN	AMS Technical Number	PCMN
M	77E	POSTDT	Posting Date	POSTDT
M	77E	POSTTM	Posting Time	POSTTM
M	77E		<CTRL><LNFD>	
M	77E	IMNTID	AMS Instrument Identifier	IMNT_ID
M	77E	ACUS	Valid Public CUSIP / CINS	ID_CUSIP
M	77E	SNAME	AMS Instrument Short Name	AMS_IMNT_SHORT_NM
M	77E	MKTSEC	AMS Market Sector Description	MARKET_SECTOR_DES
M	77E	MKTSSEC	AMS Market Sub Sector Description	MARKET_SUB_SEC_DES
M	77E	ASCT	AMS Security Instrument Type	SECURITY_TYP
M	77E	SASCT	AMS Security Instrument Sub Type	SECURITY_SUB_TYP
M	77E	MICO	AMS Accrual Calculation Type	CALC_TYP
M	77E	TIPS	TIPS Accrual Calculation Code	TIPS
M	77E	COMEXC	Composite Exchange Code	EXCH_CODE
M	77E	NAME	Name	NAME
M	77E	LNAME	Long Company Name	LONG_COMP_NAME
M	77E	AECH	Equity Primary Exchange Code	EQY_PRIM_EXCH_SHRT
M	77E	ATK	Instrument Ticker	TICKER
M	77E	DENO	Country the Instrument is issued in (ISO)	CNTRY_ISSUE_ISO
M	77E	CONSTP	Constantly Priced Instrument	CONSTANT_PR
M	77E	INCCY	Equity Dividend / Income Currency is paid	EQY_DVD_CRNCY
M	77E	ISRCTY	Country of Issuer (ISO)	COUNTRY_ISO
M	77E	AISS	Issuer Name	ISSUER
M	77E	AUNP	Par Amount	PAR_AMT
M	77E	PRCCCY	Currency Instrument is Priced in	CRNCY
M	77E	MKTMULT	Price Factor NB (price multiplier to calc MV)	PRICE_FACTOR_NB
M	77E	AUNP	Par Value Currency (principal payments)	PAR_VAL_CRNCY
M	77E	ACLAS 0,1,2, repeating	Classification, Industry codes	Initially, up to 10 codes

Sample Instrument Add Tags for Corporate, Government

IMNTADD for Corporates and Governments (public securities)

Status	Tag	SWIFT Field	Liberty Tag	AMS / LITE TAG
M	20		Transaction Ref Number: Message layer unique technical number	
M	12		Message description tag MT598	MSGID
M	77E		<CTRL><LNFD>	
M	77E	MSGTYP	Instrument Message Type	MSGTYP
M	77E	DATASRC	Data Source Code	DATASRC
M	77E	DATATRG	Data Target	DATATRG
M	77E	PCMN	AMS Technical Number	PCMN
M	77E	POSTDAT	Posting Date	POST_DAT
M	77E	POSTTIM	Posting Time	POST_TIM
M	77E		<CTRL><LNFD>	
M	77E	IMNTID	AMS Instrument Identifier	IMNT_ID
M	77E	ACUS	Valid Public CUSIP / CINS	ID_CUSIP
M	77E	SNAME	AMS Instrument Short Name	AMS_IMNT_SHORT_NM
M	77E	MKTSEC	AMS Market Sector Description	MARKET_SECTOR_DES
M	77E	MKTSSEC	AMS Market Sub Sector Description	MARKET_SUB_SEC_DES
M	77E	ASCT	AMS Security Instrument Type	SECURITY_TYP
M	77E	SASCT	AMS Security Instrument Sub Type	SECURITY_SUB_TYP
M	77E	MICO	AMS Accrual Calculation Type	CALC_TYP
M	77E	TIPS	TIPS Accrual Calculation Code	TIPS
M	77E	COMEXC	Composite Exchange Code	EXCH_CODE
M	77E	NAME	Name	NAME
M	77E	LNAME	Long Company Name	LONG_COMP_NAME
M	77E	AECH	Equity Primary Exchange Code	EQY_PRIM_EXCH_SHRT
M	77E	DENO	Country of Issue of the Instrument ISO	CNTRY_ISSUE_ISO
M	77E	CONSTP	Constantly Priced Instrument	CONSTANT_PR
M	77E	AITM	Day-count convention for calc accr'd int	DAY_CNT
M	77E	ACDS	Coupon Type	CPN_TYP
M	77E	COUP	Current coupon	CUR_CPN
M	77E	DDTE	Dated Date	INTEREST_ACC_DT
M	77E	FCPNNDT	First Coupon Date	FIRST_CPN_DT
M	77E	COUP	Coupon Currency	CPN_CRNCY
M	77E	AFQY	Coupon / Income frequency	CPN_FREQ
M	77E	ISRCTY	Country of Issuer (ISO)	COUNTRY_ISO
M	77E	AISS	Issuer Name	ISSUER
M	77E	LRSETDT	Last Rate Reset Date	LAST_REFIX_DT
M	77E	AMD	Maturity Date	MATURITY
M	77E	MOODY	Moody Rating	RTG_MOODY
M	77E	ANPD	Next Coupon Date	NXT_CPN_DT
M	77E	NXTRSET	Next Rate Reset Date	NXT_REFIX_DT
M	77E	PCPNNDT	Previous Coupon Date	PREV_CPN_DT
M	77E	AUNP	Par Amount	PAR_AMT
M	77E	PRCCCY	Pricing currency	CRNCY
M	77E	MKTMULT	Price Factor NB (price multiplier to calc MV)	PRICE_FACTOR_NB
M	77E	PPAYCCY	Par Value Currency (principal payments)	PAR_VAL_CRNCY
M	77E	REDEM	Redemption Value	REDEMP_AM
M	77E	SPRAT	SP Rating	RTG_SP
M	77E	ZCPN	Zero Coupon Marker	ZERO_CPN
M	77E	ACLAS 0,1,2, repeating	Classification, Industry codes	Initially, up to 10 codes

Sample Instrument Add Tags for Mortgages

IMNTADD for Mortgages (public securities)

Status	Tech	SWIFT Field	Liberty Data Item	AMS/LITE TAG
M	20	BUSINESS_ID	Transaction Ref Number: Message layer unique technical number	
M	12	MSG_ID	Message description tag MT598	MSG_ID
M	77E		<CTRL><LNFD>	
M	77E	MSG_TYP	Instrument Message Type	MSG_TYP
M	77E	DATA_SRC	Data Source Code	DATA_SRC
M	77E	DATA_TRGT	Data Target	DATA_TRGT
M	77E	PCMN	AMS Technical Number	PCMN
M	77E	POST_DAT	Posting Date	POST_DAT
M	77E	POST_TIM	Posting Time	POST_TIM
M	77E		<CTRL><LNFD>	
M	77E	IMNT_ID	AMS Instrument Identifier	IMNT_ID
M	77E	ACUS	Valid Public CUSIP / CINS	ID_CUSIP
M	77E	SNAME	AMS Instrument Short Name	AMS_IMNT_SHORT_NM
M	77E	MKTSEC	AMS Market Sector Description	MARKET_SECTOR_DES
M	77E	MKTSSEC	AMS Market Sub Sector Description	MARKET_SUB_SEC_DES
M	77E	ASCT	AMS Security Instrument Type	SECURITY_TYP
M	77E	SECSTYP	AMS Security Instrument Sub Type	SECURITY_SUB_TYP
M	77E	MICO	AMS Accrual Calculation Type	CALC_TYP
M	77E	TIPS	TIPS Accrual Calculation Code	TIPS
M	77E	NAME	Name	NAME
M	77E	LNAME	Long Company Name	LONG_COMP_NAME
M	77E	DENO	Country of Issue of the Instrument - ISO	CNTRY_ISSUE_ISO
M	77E	CONSTP	Constantly Priced Instrument	CONSTANT_PR
M	77E	AITM	Day count convention for calc accrued interest	DAY_CNT
M	77E	ACDS	Coupon Type	CPN_TYP
M	77E	COUP	Current Coupon	CUR_CPN
M	77E	DDTE	Dated Date	INTEREST_ACC_DT
M	77E	FCPNDT	First Coupon Date	FIRST_CPN_DT
M	77E	INCCY	Coupon Currency	CPN_CRNCY
M	77E	AFQY	Income / Coupon Frequency	CPN_FRQ
M	77E	ISRCTY	Country of Issuer (ISO)	COUNTRY_ISO
M	77E	AISS	Issuer Name	ISSUER
M	77E	LRSETDT	Last Rate Reset Date	LAST_REFIX_DT
M	77E	AMD	Maturity Date	MATURITY
M	77E	MOODY	Moody Rating	RTG_MOODY
M	77E	MCPAYRT	MTG_CUR_PAY_RT	MTG_CUR_PAY_RT
M	77E	MFPAYDT	MTG_FACTOR_PAY_DT	MTG_FACTOR_PAY_DT
M	77E	MFPDT	MTG_FIRST_PAY_DT	MTG_FIRST_PAY_DT
M	77E	MAITM	MTG_PAY_DELAY	MTG_PAY_DELAY
M	77E	CUFU	Mortgage Current Pool Factor	MTG_POOL_FACTOR
M	77E	PRFC	Mortgage Previous Pool Factor	MTG_PREV_FACTOR
M	77E	ANPD	Next Coupon Date	NXT_CPN_DT
M	77E	NXTRSET	Next Rate Reset Date	NXT_REFIX_DT
M	77E	PCPNDT	Previous Coupon Date	PREV_CPN_DT
M	77E	AUNP	Par Amount	PAR_AMT
M	77E	PRCCCCY	Currency Instrument is Priced in	CRNCY
M	77E	MKTMULT	Price Factor NB (price multiplier to calc MV)	PRICE_FACTOR_NB
M	77E	PPAYCCY	Par Value Currency (principal payments)	PAR_VAL_CRNCY
M	77E	REDEM	Redemption Value	REDEMP_AM

<i>Status</i>	<i>Tag</i>	<i>SWIFT Field</i>	<i>Library Data Item</i>	<i>ANSI / X.121 Tag</i>
M	77E	SPRAT	SP Rating	RTG_SP
M	77E	ZCPN	Zero Coupon Marker	ZERO_CPN
M	77E	ACLAS 0,1,2, repeating	Classification, Industry codes	Initially, up to 10 codes

EV478774021US

Sample Instrument Add Tags for Municipals

IMNTADD for Municipalities (public securities)

Status	Tag	Liberty Tag	Description	AMS / ILITE TAG
M	20		Transaction Ref Number: Message layer unique technical number	
M	12		Message description tag MT598	MSG_ID
M	77E		<CTRL><LNFD>	
M	77E	MSGTYP	Instrument Message Type	MSG_TYP
M	77E	DATASRC	Data Source Code	DATA_SRC
M	77E	DATATRG	Data Target	DATA_TRGT
M	77E	PCMN	AMS Technical Number	PCMN
M	77E	POSTDAT	Posting Date	POST_DAT
M	77E	POSTTIM	Posting Time	POST_TIM
M	77E		<CTRL><LNFD>	
M	77E	IMNTID	AMS Instrument Identifier	IMNT_ID
M	77E	ACUS	Valid Public CUSIP / CINS	ID_CUSIP
M	77E	SNAME	AMS Instrument Short Name	AMS_IMNT_SHORT_NM
M	77E	MKTSEC	AMS Market Sector Description	MARKET_SECTOR_DES
M	77E	MKTSSEC	AMS Market Sub Sector Description	MARKET_SUB_SEC_DES
M	77E	ASCT	AMS Security Instrument Type	SECURITY_TYP
M	77E	SECSTYP	AMS Security Instrument Sub Type	SECURITY_SUB_TYP
M	77E	MICO	AMS Accrual Calculation Type (Investigate)	CALC_TYP
M	77E	TIPS	TIPS Accrual Calculation Code	TIPS
M	77E	NAME	Name	NAME
M	77E	LNAME	Long Company Name	LNAME
M	77E	DENO	Country of Issue of the Instrument (ISO)	CNTRY_ISSUE_ISO
M	77E	CONSTP	Constantly Priced Instrument	CONSTANT_PR
M	77E	AITM	Day Count	DAY_CNT
M	77E	ACDS	Coupon Type	CPN_TYP
M	77E	COUP	Current Coupon	CUR_CPN
M	77E	DDTE	Dated Date	INTEREST_ACC_DT
M	77E	FCPNDT	First Coupon Date	FIRST_CPN_DT
M	77E	INCCY	Coupon Currency	CPN_CRNCY
M	77E	AFQY	Coupon Frequency	CPN_FRQ
M	77E	ISRCTY	Country of Issuer (ISO)	COUNTRY_ISO
M	77E	AISS	Issuer Name	ISSUER
M	77E	LRSETDT	Last Rate Reset Date	LAST_REFIX_DT
M	77E	AMD	Maturity Date	MATURITY
M	77E	MOODY	Moody Rating	RTG_MOODY
M	77E	ADV_RFND_PX	MUNI_ADV_RFND_PX	MUNI_ADV_RFND_PX
M	77E	ASET	MUNI_ADJ_CPN_MODE	MUNI_ADJ_CPN_MODE
M	77E	ISSUE_TYP	MUNI_ISSUE_TYP	MUNI_ISSUE_TYP
M	77E	MPPREP	MUNI_PRE_REFND_MTY_DT	MUNI_PRE_REFND_MTY_DT
M	77E	ANPD	Next Coupon Date	NXT_CPN_DT
M	77E	NXTRSET	Next Coupon Reset Date (NXT_REFIX_DT)	NXT_REFIX_DT
M	77E	PCPNDT	Previous Coupon Date	PREV_CPN_DT
M	77E	AUNP	Par Amount	PAR_AMT
M	77E	PRCCCY	Currency Instrument is Priced in	CRNCY
M	77E	MKTMULT	Price Factor NB (price multiplier to calc MV)	PRICE_FACTOR_NB
M	77E	AUNP	Par Value Currency (principal paymnt)	PAR_VAL_CRNCY
M	77E	REDEM	REDEMP_AM	REDEMP_AM

Status	Tag	Liberty Tag	Description	AMS/LITE TAG
M	77E	SPRAT	RTG_SP	RTG_SP
M	77E	ZCPN	ZERO_CPN	ZERO_CPN
M	77E	ASC	STATE_CODE	STATE_CODE
		ACLAS 0,1,2, repeating	Classification, Industry codes	Initially, up to 10 codes

Sample Instrument Add Tags for Private Placements

IMNTADD for Municipalities (public securities)

Status	Tag	SWIFT Field	Description	AMS/LITE TAG
M	20		Transaction Ref Number: Message layer unique technical number	
M	12		Message description tag MT598	MSG_ID
M	77E		<CTRL><LNFD>	
M	77E	MSGTYP	Instrument Message Type	MSGTYP
M	77E	DATASRC	Data Source Code	DATASRC
M	77E	DATATRG	Data Target	DATATRG
M	77E	PCMN	AMS Technical Number	PCMN
M	77E	POSTDT	Posting Date	POSTDT
M	77E	POSTTM	Posting Time	POSTTM
M	77E		<CTRL><LNFD>	
M	77E	IMNTID	AMS Instrument Identifier	IMNT_ID
M	77E	SNAME	AMS Instrument Short Name	AMS_IMNT_SHORT_NM
M	77E	MKTSEC	AMS Market Sector Description	MARKET_SECTOR_DES
M	77E	MKTSSEC	AMS Market Sub Sector Description	MARKET_SUB_SEC_DES
M	77E	ASCT	AMS Security Instrument Type	SECURITY_TYP
M	77E	MICO	AMS Accrual Calculation Type (Investigate)	CALC_TYP
M	77E	TIPS	TIPS Accrual Calculation Code	TIPS
M	77E	NAME	Name	NAME
M	77E	NDESC1	Line Description 1	Line Description 1
M	77E	NDESC2	Line Description 2	Line Description 2
M	77E	NDESC3	Line Description 3	Line Description 3
M	77E	NDESC4	Line Description 4	Line Description 4
M	77E	NDESC5	Line Description 5	Line Description 5
M	77E	AITM	Day Count	DAY_CNT
M	77E	ACDS	Coupon Type	CPN_TYP
M	77E	COUP	Current Coupon	CUR_CPN
M	77E	AMD	Maturity Date	MATURITY
M	77E	SERV	Servicer	Servicer
M	77E	CHAR	Servicer Fee Rate	CHAR
M	77E	PPNUM	Private Placement ID	PP#
M	77E	PRIADM	Prin Administrator	PRIN_ADM
M	77E	PPTYCOST		
M	77E	VAULT	Vaulted	Vault
M	77E	NOSENV	No. Sealed Envelops	NO_SEAL_ENV
M	77E	ACLAS 0,1,2, repeating	Classification, Industry codes	Initially, up to 10 codes
			Other common attributes will be identified in examples	

Sample Instrument Add Tags for Exchange Traded Derivatives

IMNTADD for Municipalities (public securities)

Status	Tag	SWIFT Field	Description	AMS/LITE TAG
M	20		Transaction Ref Number: Message layer unique technical number	

Status	Tag	SWIFT Field	Description	AMS/LITE TAG
M	12		Message description tag MT598	MSG_ID
M	77E		<CTRL><LNFD>	
M	77E	MSGTYP	Instrument Message Type	MSG_TYP
M	77E	DATASRC	Data Source Code	DATA_SRC
M	77E	DATATRG	Data Target	DATA_TRGT
M	77E	PCMN	AMS Technical Number	PCMN
M	77E	POSTDT	Posting Date	POST_DAT
M	77E	POSTTM	Posting Time	POST_TIM
M	77E		<CTRL><LNFD>	
M	77E	IMNTID	AMS Instrument Identifier	IMNT_ID
M	77E	ACUS	Valid Public CUSIP / CINS	ID_CUSIP
M	77E	SNAME	AMS Instrument Short Name	AMS_IMNT_SHORT_NM
M	77E	MKTSEC	AMS Market Sector Description	MARKET_SECTOR_DES
M	77E	ASCT	AMS Security Instrument Type	SECURITY_TYP
M	77E	MICO	AMS Accrual Calculation Type	CALC_TYP
M	77E	TIPS	TIPS Accrual Calculation Code	TIPS
M	77E	NAME	Name	NAME
M	77E	LNAME	Long Company Name	LNAME
M	77E	AECH	Equity Primary Exchange Code	EQY_PRIM_EXCH_SHRT
M	77E	CNTRY_ISSUE	Country of Issue of the Instrument (ISO)	CNTRY_ISSUE_ISO
M	77E	CONSTANT_PR	Constantly Priced Instrument	CONSTANT_PR
M	77E	CNTRY_ISSUER	Country of Issuer (ISO)	COUNTRY_ISO
M	77E	AISS	Issuer Name	ISSUER
M	77E	EXPDT	Expiry Date	EXPIRY_DT
M	77E	CTRMLT	Contract Multiplier	CONTRACT_MULT
M	77E	LINKID	AMS Underlying Instrument	Linked Instrument ID
M	77E	ACLAS 0,1,2, repeating	Classification, Industry codes	Initially, up to 10 codes

IMNTMOD

AMS will message to BNY changes to attributes for existing instruments using an MT598 message, with a specific message type tag defining the message as a change of attributes to an existing instrument. AMS will transmit only the attributes that have changed. BNY's response to IMNTMOD will be an IMNTACK message, containing the full SMDB data record for the security.

Status	Tag	SWIFT Field	Liberty Data Item	AMS/LITE TAG
M	20		Transaction Ref Number: Message layer unique technical number	
M	12		Message description tag MT598	MSG_ID
M	77E		<CTRL><LNFD>	
M	77E	MSG_TYP	Instrument Message Type	MSG_TYP
M	77E	DATA_SRC	Data Source Code	DATA_SRC
M	77E	DATA_TRGT	Data Target	DATA_TRGT
M	77E	PCMN	AMS Technical Number	PCMN
M	77E	POST_DAT	Posting Date	POST_DAT
M	77E	POST_TIM	Posting Time	POST_TIM
M	77E		<CTRL><LNFD>	
M	77E	Changed attributes	Changed attributes	Changed attributes
M	77E	IMNT_ID	AMS Instrument Identifier	IMNT_ID
M	77E	INSM_NO	AMS Instrument Number	INSM_NO

IMNTSUB

Assets AMS sends to BNY using IMNTADD will be added to the BNY SMDB subscription list automatically. AMS systems will subscribe to SMDB for pricing data schedules, and corporate action information. The IMNTACK message SMDB sends to AMS acts a confirmation that the asset is added to the subscription list.

AMS can subscribe or un-subscribe to assets (A: Add, D: Delete) outside of using the IMNTADD function for pricing, corporate actions and indicative data. AMS will use IMNTSUB also to request an asset refresh.

The use of IMNTSUB for pricing and corporate actions is not included in phase 1, and out of scope for this version of the document. This functionality will be detailed later under separate cover.

Status	Tag	SWIFT Field	Liberty Data Item	AMS / LITE TAG
M	20		Transaction Ref Number: Message layer unique technical number	
M	12		Message description tag MT598	MSG_ID
M	77E		<CTRL><LNFD>	
M	77E	MSG_TYP	Instrument Message Type	MSG_TYP
M	77E	DATA_SRC	Data Source Code	DATA_SRC
M	77E	DATA_TRGT	Data Target	DATA_TRGT
M	77E	PCMN	AMS Technical Number	PCMN
M	77E	POST_DAT	Posting Date	POST_DAT
M	77E	POST_TIM	Posting Time	POST_TIM
M	77E	ACTION_FLAG	Subscription action (A: Add, D: Delete)	ACTION_FLAG
M	77E	DATA_REQ	Subscription data of Interest (P: Price, C: CA, S: Schedules, I: Indicative)	DATA_REQ

IMNTREC

For phase 1, reconciliation will be handled via a daily download sent from AMS to SMDB. Security information that differs from information that exists on SMDB is added to the SMDB break-queue mechanism. The attributes covered by IMNTREC are defined in the attributes spreadsheet.

Status	Tag	SWIFT Field	Liberty Data Item	AMS / LITE TAG
M	20		Transaction Ref Number: Message layer unique technical number	
M	12		Message description tag MT598	MSG_ID
M	77E		<CTRL><LNFD>	
M	77E	MSG_TYP	Instrument Message Type	MSG_TYP
M	77E	DATA_SRC	Data Source Code	DATA_SRC
M	77E	DATA_TRGT	Data Target	DATA_TRGT
M	77E	PCMN	AMS Technical Number	PCMN
M	77E	POST_DAT	Posting Date	POST_DAT
M	77E	POST_TIM	Posting Time	POST_TIM
M	77E	REGION	Regional subscription and time zone information (NY, LN, TO)	REGION

IMNTACK

BNY SMDB will respond to MNTADD and IMNTMOD messages with an IMNTACK message. The IMNTACK is a positive acknowledgement message and will include the full SMDB data record for the security.

Status	Tag	SWIFT Field	Liberty Data Item	AMIS ALIVE TAG
M	20		Transaction Ref Number: Message layer unique technical number	
M	12		Message description tag MT598	MSG_ID
M	77E		<CTRL><LNFD>	
M	77E	MSG_TYP	Instrument Message Type	MSG_TYP
			All data attributes...	

The full list of SWIFT-like tags that will be returned in the IMNTACK message can be found under separate cover in the SMDB Tags spreadsheet.

For break-queue additions, based on an AMS initiated IMNTADD or IMNTMOD message, SMDB will respond with an IMNTACK when the break item is resolved. Also, SMDB will respond with an IMNTACK message for AMS initiated IMNTSUB messages.

Data Translation and Inclusion in IMNTACK

Data will be translated into the SWIFT message format by SMDB when it is returned to AMS. Attributes received by SMDB from AMS will be treated three different ways:

- Attributes received and not reflected back.
- Attributes received are translated into SMDB on the inbound and translated to SWIFT mnemonics, followed SMDB tags on the outbound to AMS.
- SMDB direct mapping of attributes and reflected back to AMS.

Note: The list of fields being translated, fields not being reflected back, and fields being mapped one for one will have been agreed to, and can be found under separate cover in the attributes spreadsheet.

Exception items and other items of note are listed below.

- For every AMS identifier, there will be only one corresponding SMDB identifier.
- JPM will internally map each of the Bloomberg codes they use for accrual calculation to TIPS accrual routines. AMS will send the agreed upon code that relates to a specific TIPS routine, along with a day count to SMDB. Both AMS and BNY will translate to internal accounting systems using these codes. AMS does not expect these attributed returned.
- Market sector, security type and coupon type attributes will be transmitted to BNY, and will be used to generate the BNY internal asset classification. AMS does not expect SMDB to reflect these attributes back.
- AMS will transmit up to ten asset classification schemes, which SMDB will carry and reflect back to AMS in the IMNTACK message. Placeholders are defined in this document for asset classification. AMS reporting requirements will drive the schemes SMDB will need to carry on their local database, and will provide these schemes to SMDB.

To populate the asset structures, AMS will send certain coded fields that are unique to AMS to SMDB in IMNTADD messages. These fields will not be modified by SMDB, and will be returned to AMS in the same format as they were received. In

addition, these fields will be mapped to CDW so that when responsibility for JPM client reporting is transferred to CDW, they will be able to accommodate this requirement (definition and translation of the various codes will be required at some point).

Note: Additional detail regarding the transmission of asset classification codes and the explanation of these codes will be provided under separate cover.

- The SMDB rules that generate the descriptions have been reviewed by AMS and BNY Business Operations, and have been approved for phase 1. Changes to these rules may be needed to meet AMS Private Client Group requirements.
- There will be no distinction between held securities and research securities in IMNTADD messages. However, work queue items related to held securities must be addressed before work queue items that relate to research securities. Therefore, the GSP work queue, which will only contain trade related, and therefore held, securities, will be cross-referenced. Those break items that are found on the GSP work queue and the AMS work queue will be handled first.

IMNTNAK

Technical negative acknowledgments will be sent from BNY to AMS for messages that fail validation performed by the BNY middleware product, MIDYS.

IMNTLST

Reference tables that support the transformation of data between AMS and SMDB will be synchronized between the two databases. The process by which the synchronization will be accomplished is detailed below.

Either AMS or SMDB will be identified as the owner of the specific reference table. When the owner of the table identifies a change, notification will be sent to the opposite database. The new reference table will be sent, with the old and new contents defined. The non-owning database will run a 'Dif' program to identify the difference between the two and the new information will then overlay the old.

Note: The mode of notifying the non-owning database of a new reference value will be determined jointly with the two data management teams.

The IMNTLST message will support the synchronization of these tables. The list of tables to be contained in these messages is as follows:

Ref Value	Description
Calendar	Global Holiday Calendar
Currency, Country	Currency and Country
Exchanges	Exchange
Classifications	Classification Codes, Structures and Relationships

Notes:

- Country and currency codes are ISO standard.
- Reference tables regarding prices and corporate actions will be identified in their respective specification documents.

Status	Tag	SWIFT Field	Open Data Item	AMSLITE Tag

Status	Tag	SWIFT Field	Liberty Data Item	AMSLITE TAG
M	20		Transaction Ref Number: Message layer unique technical number	
M	12		Message description tag MT598	MSG_ID
M	77E		<CTRL><LNFD>	
M	77E	MSG_TYP	Instrument Message Type	MSG_TYP
M	77E	DATA_SRC	Data Source Code	DATA_SRC
M	77E	DATA_TRGT	Data Target	DATA_TRGT
M	77E	PCMN	AMS Technical Number	PCMN
M	77E	POST_DAT	Posting Date	POST_DAT
M	77E	POST_TIM	Posting Time	POST_TIM
M	77E	REGION	Regional subscription and time zone information (NY, LN, TO)	REGION
M	77E	REF_TBL	Reference Table to update	REF_TBL
M	77E	ATTRIB_BEGIN		
M	77E	ATTRIB_END		

Note: The attributes in this message will be dependent on the tables being sent in the message.

IMNTPRICEPUSH and IMNTCAPUSH

Price messages and corporate action event messages will be sent to AMS on a price and corporate action subscription basis, respectively. The content of these messages is not covered by the scope of this document and will be detailed later under separate cover.

Note: An indicative subscription to a security is a pre-requisite to establishing a price or corporate action subscription.

Sample Messages

IMNTADD Equity

```
{4:  
:20:123456  
:12:598  
<CTRL><LNFD>  
:77E:/MSG_TYP/IMNTADD  
/DATA_SRC/ILITE  
/DATA_TRGT/SMDB  
/PCMN/123456  
/POST_DAT/19990915  
/POST_TIM/17:30:25EST  
<CTRL><LNFD>  
/IMNT_ID/100001  
/INSM_NO/90000122  
/ACUS/013817101  
/AMS_IMNT_SN/ALCOA INC  
/AMS_MKT_SEC/Equity  
/ASCT/Common Stock  
/MICO/001  
/COMP_CODE/US  
/NAME/Alcoa Inc.  
/LNAME/Alcoa Inc.  
/AECH/UN  
/ATK/AA  
/CNTRY_ISSUE/US  
/EQY_DVD_CRNCY/USD  
/CNTRY_ISSUER/US  
/AISS/Alcoa Inc.  
/PRC_CUR/USD  
/PRC_MULT_MV/1.0  
-}
```

IMNTADD Govt

```
{4:  
:20:123456  
:12:598  
<CTRL><LNFD>  
:77E:/MSG_TYP/IMNTADD  
/DATA_SRC/ILITE  
/DATA_TRGT/SMDB  
/PCMN/123456  
/POST_DAT/19990915  
/POST_TIM/17:30:25EST  
<CTRL><LNFD>  
/IMNT_ID/100001  
/INSM_NO/90000122  
/ACUS/Q08184EW1  
/AMS_IMNT_SN/AUSTLA13 ½ 07/10/99  
/AMS_MKT_SEC/GOVT  
/ASCT/BULLDOG  
/MICO/001  
/COMP_EXCH/LN  
/NAME/AUSTLA13 ½ 07/10/99  
/LNAME/ AUSTLA13 ½ 07/10/99  
/AECH/LN
```

```

/CNTRY_ISSUE/GB
/CONSTANT_PR/
/AITM/02
/CPN_TYP/FIXED
/COUP/13.5
/DDTE/19990128
/FIRST_CPN_DT/19990728
/CPN_CUR/GBP
/AFQY/02
/CNTRY_ISSUER/AU
/AISS/ AUSTLA13 ½ 0710/99
/AMD/20100728
/MOODY_RATING/Aa3
/NXT_CPN_DT/19990728
/PRC_CUR/GBP
/PRC_MULT_MV/0.01
/PRI_PAY_CUR/GBP
/REDEMP_AM/50000000
/SP_RATING/A+
-}

```

IMNTADD MBS

```

{4:
:20:123456
:12:598
<CTRL><LNFD>
:77E:/MSG_TYP/IMNTADD
/DATA_SRC/ILITE
/DATA_TRGT/SMDB
/PCMN/123456
/POST_DAT/19990915
/POST_TIM/17:30:25EST
<CTRL><LNFD>
/IMNT_ID/100003
/INSM_NO/90000124
/ACUS/31366HTZ6
/AMS_IMNT_SHORT_NM/FN 149168
/AMS_MKT_SEC_DES/MTGE
/AMS_MKT_SUB_SEC_DES/
/ASCT/MBS Other
/MICO/001
/NAME/FN 149168
/LONG_COMP_NAME/FANNIE MAE
/CNTRY_ISSUE/US
/AITM/03
/CPN_TYP/FIXED
/COUP/10
/DDTE/19900401
/FIRST_CPN_DT/19900501
/CPN_CRNCY/USD
/AFQY/06
/COUNTRY_ISO/US
/AIIS/FANNIE MAE
/AMD/20200401
/MOODY_RATING/Aaa
/CUR_PAY_RT/10
/FACTOR_PAY_DT/19990415
/FIRST_PAY_DT/19900515
/PAY_DELAY/15
/CUFU/.892536
/PRFC/.898672

```

```

/NXT_CPN_DT/19990515
/PREV_CPN_DT/19990315
/PRC_CUR/USD
/PRC_MULT_MV/0.01
/PRI_PAY_CUR/USD
/REDEMP_AM/50000000
/SP_RATING/AAA
-}

```

IMNTADD MUNI

```

{4:
:20:123456
:12:598
<CTRL><LNFD>
:77E:/MSG_TYP/IMNTADD
/DATA_SRC/ILITE
/DATA_TRGT/SMDB
/PCMN/123456
/POST_DAT/19990915
/POST_TIM/17:30:25EST
<CTRL><LNFD>
/IMNT_ID/100004
/INSM_NO/90000125
/ACUS/008350AH3
/AMS_IMNT_SHORT_NM/AFTON CENT SCH
/AMS_MKT_SEC/Muni
/ASCT/Bond
/MICO/001
/NAME/AFTON CENT SCH
/LONG_COMP_NAME/AFTON CENT SCH
/CNTRY_ISSUE/US
/AITM/01
/CPN_TYP/FIXED
/COUP/5.1
/DDTE/19890615
/FIRST_CPN_DT/19900615
/CPN_CUR/USD
/AFQY/01
/CNTRY_ISSUER/US
/AISS/AFTON CENT SCH D
/AMD/19990615
/MOODY_RATING/Baa1
/ISSUE_TYP/GENERAL OBLIGATION UNLTD
/NXT_CPN_DT/19990615
/FRNR/19980615
/PREV_CPN_DT
/PRC_CUR/USD/USD
/PRC_MULT_MV/0.01
/PAR_VAL_CRNCY/USD
/REDEMP_AM/50000000
/SP_RATING/BBB
/ASC/NY

```

Data Field	Tag Num	SMDB Column	SMDB Table	Comments
SMDSELCT				EV478774021US
API TAG FI BEGIN	^822^			
API TAG CLASS	^123^	CLAS_TX	T_SMD_FI	
API TAG COLLATERAL_TYPE	^125^	COLAT_TYP_CD	T_SMD_FI	
API TAG CONTRACT_LOT_SIZ	^129^	CNTR_LOT_SIZE_AM	T_SMD_FI	
API TAG LGLEN_COUNTRY_INCORP	^135^	CTRY_INC_CD	T_SMD_LGLEN	
API TAG COUNTRY_OF_COLL	^136^	CTRY_COLAT_CD	T_SMD_FI	
API TAG FI CURRENT RATE	^142^	CUR_RT	T_SMD_FI	
API TAG INCOME_CURRENCY	^162^	PRM_CCY_CD	T_SMD_FI	
API TAG EXPIRATION_DATE	^305^	EXP_DT	T_SMD_FI	
API TAG ANTIC_Maturity_DATE	^308^	ANTC_MAT_DT	T_SMD_FI	
API TAG FI FIRST_CPN_DATE	^309^	FST_CPN_DT	T_SMD_FI	
API TAG FIRST_RESET_DATE	^310^	FST_RST_DT	T_SMD_FI	
API TAG INCREMENTAL_FUNGIBILITY	^316^	INCR_FNG_AM	T_SMD_FI	
API TAG INSM_NO	^318^	INSM_NO	T_SMD_FI	SMDB instrument number
API TAG INSM_TYPE_CODE	^336^		T_SMD_FI	
API TAG FI SHORT_DESC	^G1E^	BNY_SHRT_DESC_TX	T_SMD_FI	
API TAG FI DESC1	^G11^	BNY_DESC_LINE1_TX	T_SMD_FI	
API TAG FI DESC2	^G12^	BNY_DESC_LINE2_TX	T_SMD_FI	
API TAG FI DESC3	^G13^	BNY_DESC_LINE3_TX	T_SMD_FI	
API TAG FI DESC4	^G14^	BNY_DESC_LINE4_TX	T_SMD_FI	
API TAG FI DESC5	^G15^	BNY_DESC_LINE5_TX	T_SMD_FI	
API TAG FI ISSUE_DATE	^334^	ISUE_DT	T_SMD_FI	
API TAG ISSUER_NAME	^402^	PRM_NM	T_SMD_LGLEN	
API TAG ISSUING_CURRENCY	^403^	ISUE_CCY_CD	T_SMD_FI	
API TAG ISSUING_COUNTRY	^404^	ISUE_CTR_CD	T_SMD_FI	
API TAG MINIMUM_FUNGIBILITY	^444^	MIN_FNG_AM	T_SMD_FI	
API TAG FI DAY_IN_PERIOD_CT	^512^	DY_IN_PER_CT	T_SMD_FI	
API TAG FI DAY_IN_YEAR_CT	^515^	DY_IN_YR_CT	T_SMD_FI	
API TAG OPTION_TYPE_CODE	^543^	OPT_TYP_CD	T_SMD_FI	
API TAG OID_PRICE	^550^	OID_PRC_AM	T_SMD_FI	
API TAG ORIG_MAT_DATE	^561^	ORGN_MAT_DT	T_SMD_FI	
API TAG FI_PAR_VALUE	^565^	PAR_VAL_AM	T_SMD_FI	
API TAG PAYMENT_FREQUENCY	^600^	PMT_FQ_CD	T_SMD_FI	
API TAG PAYMENT_DAY_NO	^601^	PMT_DY_NO	T_SMD_FI	
API TAG PRINCIPAL_CURRENCY	^651^	PRIN_CCY_CD	T_SMD_FI	
API TAG RESET_CAP	^656^	CAP_TERM_CT	T_SMD_FI	
API TAG RESET_FLOOR	^657^	FLR_TERM_CT	T_SMD_FI	
API TAG SIC_CODE	^673^			
API TAG STATE_CODE	^674^	ST_CD	T_SMD_FI	
API TAG STRIKE_PRICE	^676^	STRK_PRC_AM	T_SMD_FI	
API TAG TICKER	^782^	FI_NBR_SYS_NO	T_SMD_FI_NBR_SHM	
API TAG WORKFLOW_ID	^787^			
API TAG TRANCHE_NUMBER	^789^	TRNCH_NO	T_SMD_FI	
API TAG REL_TICKER	^790^	FI_NBR_SYS_NO	T_SMD_FI_NBR_SHM	
API TAG ISR_NO	^797^	ISR_NO	T_SMD_FI	
API TAG ORIG_ISUE_AMOUNT	^798^	ORGN_ISUE_AM	T_SMD_FI	
API TAG ISUE_OUTS_AMOUNT	^799^	ISUE_OUTS_AM	T_SMD_FI	
API TAG ISUE_OUTS_EFF_DATE	^800^	ISUE_OUTS_EFF_DT	T_SMD_FI	
API TAG TOTAL_SHARE_AMT	^801^	TOT_SHR_AM	T_SMD_FI	
API TAG DELAY_DAY_COUNT	^802^	DELY_DY_CT	T_SMD_FI	
API TAG INT_FROM_DATE	^803^	INT_FROM_DT	T_SMD_FI	
API TAG MULT_RATE	^500^	MULT_RT	T_SMD_FI	
API TAG UNIT_MULT_RATE	^804^	UNIT_MULT_RT	T_SMD_FI	
API TAG POOL_PREFIX_CD	^805^	POOL_PRFX_CD	T_SMD_FI	
API TAG NEXT_COUPON_DATE	^806^			
API TAG REVENUE_COUNTRY	^811^			
API TAG PRIMARY_TRADING_COUNT	^812^			
API TAG PRIMARY_TRADING_CURRE	^813^			

Data Field	Tag Num	SMDB Column	SMDB Table	Comments
API_TAG_EXPOSURE_CURRENCY	^814^			
API_TAG_BNY_TYPE_CODE	^G01^		T_SMD_FI	
API_TAG_FI_STATUS_CODE	^G16^	STAT_CD	T_SMD_FI	
API_TAG_RECORD_END	^END^			

API_TAG_FI_CHAR_BEGIN	^823^	-	Repeating Group
API_TAG_CHAR_TYP	^100^	TYP_CD	T_SMD_FI_CHAR
API_TAG_RECORD_END	^END^		

API_TAG_RATINGS_BEGIN	^824^	-		
API_TAG_MOODYS_RATING	^160^	CR_RTNG_CLAS_CD	T_SMD_FI_CR_RTNG	In Credit Rating Table, Not FI
API_TAG_SNP_RATING	^163^	CR_RTNG_CLAS_CD	T_SMD_FI_CR_RTNG	Not FI
API_TAG_RATING_NO	^R01^	CR_RTNG_NO	T_SMD_FI_CR_RTNG	Value i.e. AAA, aaB
API_TAG_EFF_DT	^820^	EFF_DT	T_SMD_FI_NBR_SHM	Format yyyy-mm-dd Should always be present.
API_TAG_END_DT	^821^	END_DT	T_SMD_FI_NBR_SHM	Format yyyy-mm-dd If there is no end date, tag may be missing.
API_TAG_RATING_CLASS	^R02^	CR_RTNG_CLAS_CD	T_SMD_FI_CR_RTNG	Moody rating code is '4' ' S&P rating code is '2' or '3'.
API_TAG_RECORD_END	^END^			

API_TAG_DEP_BEGIN	^825^	-		
API_TAG_DEPOSITORY_ID	^G30^	DEP_NO	T_SMD_FI_DEP_ELG	
API_TAG_EFF_DT	^820^	EFF_DT	T_SMD_FI_NBR_SHM	Format yyyy-mm-dd Should always be present.
API_TAG_END_DT	^821^	END_DT	T_SMD_FI_NBR_SHM	Format yyyy-mm-dd If there is no end date, tag may be missing.
API_TAG_INCR_LOT_SIZE	^G31^	???		CNTR_LOT_SIZE_AM is already mapped. It is not in the FI table.
API_TAG_PREF_DEP	^G32^	PREF_DEP_FL	T_SMD_FI_DEP_ELG	"y" equals preferred depository
API_TAG_RECORD_END	^END^			

API_TAG_FI_CLASS_CMP	^827^	-		
API_TAG_INSM_NO	^318^	INSM_NO	T_SMD_FI	SMDB instrument number
API_TAG_LGLEN_NO	^H01^	LGLEN_NO	T_SMD_FI_CLAS_CMP	
API_TAG_ROLE_CODE	^H02^	ROLE_CD	T_SMD_FI_CLAS_CMP	
API_TAG_CLASS_NUMBER	^H03^	CLAS_NO	T_SMD_FI_CLAS_CMP	
API_TAG_CAT_CODE	^H04^	CAT_CPN_CD	T_SMD_FI_CLAS_CMP	
API_TAG_RECORD_END	^END^			

API_TAG_REPEAT_SECID_BEGIN	^791^		Begins numbering schema grouping
----------------------------	-------	--	-------------------------------------

Data Field	Tag Num	SMDB Column	SMDB Table	Comments
API TAG SECID BEGIN	^793^	-		The group from ^793^ to ^END^ is a repeating group and occurs one to many times
API TAG SECID_COUNTRY	^795^	CTRY_CD	T_SMD FI NBR SHM	
API TAG NBR_SHM_TYP_CD	^817^	NBR_SHM_TYP_CD	T_SMD FI NBR SHM	
API TAG FI_NBR_SYS_NO	^818^	FI_NBR_SYS_NO	T_SMD FI NBR SHM	
API TAG NBR_SHM_PURP_CD	^819^	NBR_SHM_PURP_CD	T_SMD FI NBR SHM	
API TAG EFF_DT	^820^	EFF_DT	T_SMD FI NBR SHM	Format yyyy-mm-dd Should always be present.
API TAG END_DT	^821^	END_DT	T_SMD FI NBR SHM	Format yyyy-mm-dd If there is no end date, tag may be missing.
To Get ISIN				^817^ equals '0' (industry) or '41' (user assigned) Value found in ^818^
To Get SEDOL				^817^ equals '2' (industry) or '43' (user assigned) Value found in ^818^
To Get Common Code				^817^ equals '3' Value found in ^818^
To Get CUSIP				^817^ equals '1' (industry) or '44' (user assigned) Value found in ^818^
To Get Local Code				^817^ equals '52' Value found in ^818^
API TAG RECORD_END	^END^			
API TAG REPEAT_SECID_END	^792^			Ends numbering schema repeating group

API TAG REL_REPEAT_SECID_BEGIN	^691^			Begins related numbering schema grouping
API TAG REL_SECID_BEGIN	^693^			Begins related number
API TAG RELSECID_BEGIN	^826^			The group from ^693^ to ^END^ is a repeating group and occurs one to many times
API TAG SECID_COUNTRY	^795^	CTRY_CD	T_SMD FI NBR SHM	
API TAG NBR_SHM_TYP_CD	^817^	NBR_SHM_TYP_CD	T_SMD FI NBR SHM	
API TAG FI_NBR_SYS_NO	^818^	FI_NBR_SYS_NO	T_SMD FI NBR SHM	
API TAG NBR_SHM_PURP_CD	^819^	NBR_SHM_PURP_CD	T_SMD FI NBR SHM	
API TAG EFF_DT	^820^	EFF_DT	T_SMD FI NBR SHM	Format yyyy-mm-dd Should always be present.
API TAG END_DT	^821^	END_DT	T_SMD FI NBR SHM	Format yyyy-mm-dd If there is no end date, tag may be missing.
To Get related ISIN				^817^ equals '0' (industry) or '41' (user assigned) Value found in ^818^
To Get related SEDOL				^817^ equals '2' (industry) or '43' (user assigned) Value found in ^818^
To Get related Common Code				^817^ equals '3' Value found in ^818^

Data Field	Tag Num	SMDB Column	SMDB Table	Comments
To Get related CUSIP				^817^ equals '1' (industry) or '44' (user assigned) Value found in ^818^
To Get related Local Code				^817^ equals '52' Value found in ^818^
API TAG RECORD END	^END^			

API TAG_REL_INSM_BEGIN	^849^			defines relationship
API_TAG_INSM_NO	^318^	INSM_NO	T_SMD_FI	SMDB instrument number
API_TAG_REL_INSM_TYP_CD	^850^	TYP_CD	T_SMD_REL_INSM	
API_TAG_REL_INSM_EFF_DT	^851^	EFF_DT	T_SMD_REL_INSM	
API_TAG_REL_INSM_RT	^852^	RT	T_SMD_REL_INSM	
API_TAG_REL_INSM_END_DT	^853^	END_DT	T_SMD_REL_INSM	
API_TAG_REL_ISM_NO	^335^			SMDB related instrument number
API TAG RECORD END	^END^			

REPEATING GROUP

SMDB TAG NAME	SMDB TAG	SWIFT MNEMONIC	Description
AP_OBJ_SMDBNOTIFY	"SMDNOTIFY"	IMNTNOTIFY	Opening R
APPL_JPM	"^JPM^"		Feed to JPI
	"^ROU^"		Tag Group
API_TAG_EFF_DT	"^820^"	EFFDT	Effective D:
API_TAG_PR_MARKER		MSG-TYP	End of Cyc
API_TAG_PR_CYCLE		MARKERID	cycle id 20
API_TAG_NUM_OF_PRICES		NUM-OF-PRICES	Number of
API_TAG_RECORD_END	"^END^"	END	End Of Rec
API_TAG_EOM	"^EOM^"	-}	End Of Mes

{1:F01IRVTUS3NXXXX0000000001}{2:O5981713000606MGTCGB22AXX00000000000006061713N

:20:061713310884

:12:598

:77E:

-}

{1:F01IRVTUS3NXXXX0000000001}{2:O5981713000606MGTCGB22AXX00000000000006061713N

:20:061713310884

:12:598

:77E:

/MSG-TYP/MARKER

/DATA-SRC/ILITE

/DATA-TRGT/SMDB

/POST-DAT/19991029

/POST-TIM/16:05EST

/MARKERID/END-OF-DAY

/ NUM-OF-PRICES/700

-}

n

ecord Type Indicator

M

Delimiter

ate

le Marker

11,2045,2012,2046

Securities priced within the cycle

cord Delimiter

ssage Delimiter

l}{3:{113:9601}{108:com.jpmorgan.ams.xtest.servlet.Translator@1dad0db4 id 350}}{4:

l}{3:{113:9601}{108:com.jpmorgan.ams.xtest.servlet.Translator@1dad0db4 id 350}}{4:

2

ST. LOUIS, MISSOURI
WASHINGTON, D.C.
KANSAS CITY, MISSOURI
OVERLAND PARK, KANSAS
PHOENIX, ARIZONA
SANTA MONICA, CALIFORNIA
IRVINE, CALIFORNIA

BRYAN CAVE LLP

245 PARK AVENUE
NEW YORK, NY 10167-0034
(212) 692-1800
FACSIMILE: (212) 692-1900

LONDON, ENGLAND
RIYADH, SAUDI ARABIA
KUWAIT CITY, KUWAIT
ABU DHABI, UNITED ARAB EMIRATES
DUBAI, UNITED ARAB EMIRATES
HONG KONG
ASSOCIATED OFFICE IN SHANGHAI

Facsimile Delivery Instructions

This facsimile contains information which (a) may be LEGALLY PRIVILEGED, PROPRIETARY IN NATURE, OR OTHERWISE PROTECTED BY LAW FROM DISCLOSURE and (b) is intended only for the use of the Addressee(s) named below. If you are not the Addressee, or the person responsible for delivering this to the Addressee(s), you are hereby notified that reading, copying or distributing this facsimile is prohibited. If you have received this facsimile in error, please telephone us immediately and mail the facsimile back to us at the above address. Thank You.

FROM: **Bosco Kim**

DATE: May 8, 2001

MATTER NO: 123357

LAN I.D. BK2

DIRECT DIAL: (212) 692-1896

MESSAGE FROM SENDER TOTAL NO. OF PAGES: 15 (including this page)

Kirk,

Per our telephone conversation, enclosed is a copy of the draft version of the patent application for SGA.

Regards,

Bosco

PLEASE DELIVER TO THE FOLLOWING:

TO: Kirk Woetzel
COMPANY: Bank of New York

FAX NO: (212) 815-5030
PHONE NO:

TO SENDER:

Do you wish to have your copy returned?

Yes No

Do you wish to be contacted when Fax is Sent?

Yes No

Do you wish to have the confirmation sheet sent to you?

Yes No

Do you wish to be contacted at your home/office if fax **CANNOT** be sent within one hour? Phone No: 1896

Yes No

269398

If you do not receive all material, please call (212) 692-1941, or (212) 692-1913